Unisoc Coolwatcher

User Guide

Release Date	2022/06/30
Document No.	
Version	V0.3
Document Type	Word
Platform	8910/8909L. etc.
OS Version	Windows XP , Win7 , Win10, unbuntu16.04

Statement

Keywords

Environment configuration, burning, buffer, register, GDB, offline analysis, heap report, chip control, command line operation, blue screen dump.

Revision history

Version	Date	Author	Description
V0.1	**2020-06-24**	Tool-Group	**Initial Version**
V0.2	**2020-07-20**	Tool-Group	**Modify Chapter 3 and 4**
V0.3	**2022-06-30**	Tool-Group	**Add 4.13**

Preface

1. Scope

This paper introduces the basic functions and usage of the tool in detail, which is suitable for all users.

2. Details Definitions

1. Definitions
2. Symbols
3. Abbreviations

None.

3. References

None.

Contents

1. Overview

1.1 Summary

This article provides details on how to use the Coolwatcher, include hardware configuration, software configuration, steps to use and fault handing.

1.2 Goal Setting

The users can use the tool skillfully and correctly, so as to analyze and solve problems.

2. Environment

2.1 Software installation

Coolwatcher is green software, which can be used by extracting it to a subdirectory. Pay attention to the path, which cannot contain Chinese characters.

2.2 Running Environment

It is recommended that the computer be equipped with more than 4G memory and more than dual core processor.

It is necessary to ensure that the corresponding serial port driver is installed to ensure the normal communication between the module / mobile phone and the PC.

It supports Windows XP / 7 / 10, etc. But Linux only support ubuntu16.04.

2.3 License

Refer to notice.txt file.

3. Install and Setting

Coolwatcher have two different program, coolwatcher_debughost.exe and coolwatcher_usb.exe are different in communication mode.

Communication mode of coolwatcher_debughost.exe is debughost, the module and PC should be connect by serial port transfer module and port driver need to be installed first.

Coolwatcher_usb.exe use usb to communicate with PC, module and PC is connect by USB cable, usb driver is needed. Notice that some module do not support this communication mode.

3.1 Debughost Mode

3.1.1 Windows

Following will introduce 8910 usb serial port transfer module, ftdi, as an example.

3.1.1.1 Driver Installation

Install the serial port driver according to different usb serial port module.

Connect PC and module by usb serial port transfer module/line, if the transfer module is ftdi module, then install driver in <drivers\ftdi\2.12.16> folder.

After driver installed, COM number can be checked through < PC> —> <Manager> —> <Device manager> —> <ports (COM and LPT)>.

Figure 3-1-1 COMs

3.1.1.2 Check COM

How to judge which COM is connect with coolwatcher? It is always the second COM, if this method is fail, then the <hardware id> should be checked.

<Device manager> —> <ports (COM and LPT)> —> <USB serial Port(COM**)>, then right click the port and look up its property. Open the dialog of property, then click detailed information choose the < hardware id> to check the id information.

Figure 3-1-2 right click to check property

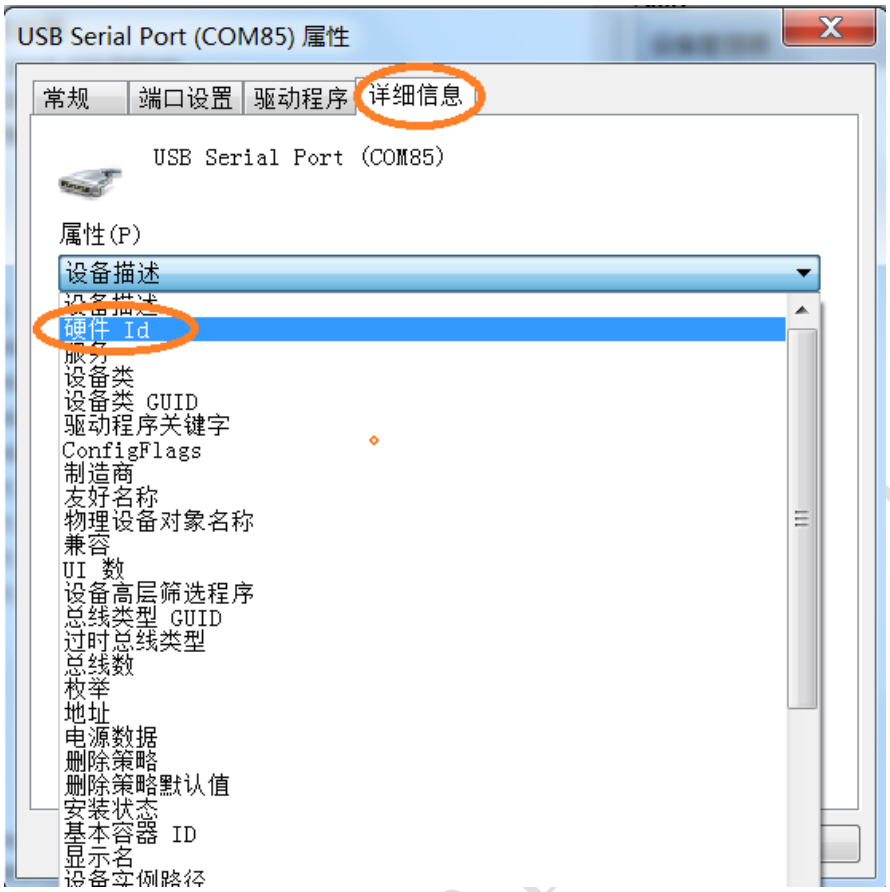


Figure 3-1-3 check hardware id - I

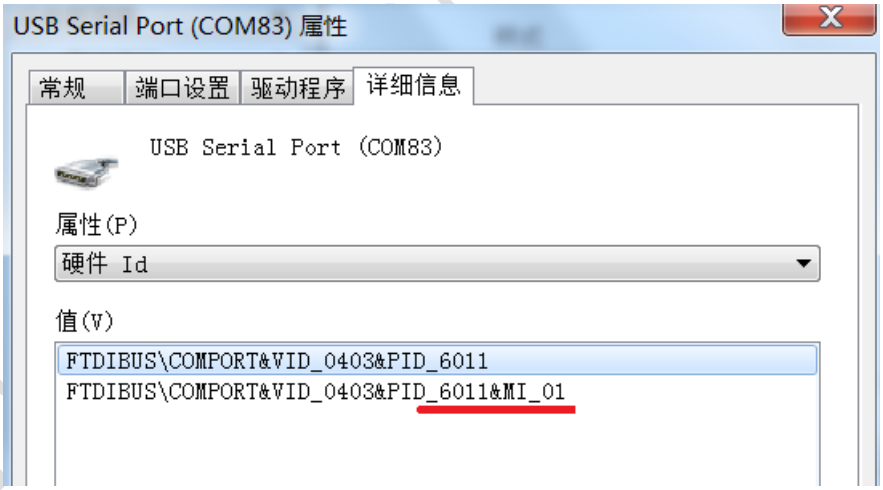


Figure 3-1-4 check hardware id - II

COM with hardware information shown in Figure 3-1-4 is the correct com that communicate with coolwatcher.

3.1.2 Ubuntu16.04

3.1.2.1 Install Dependent Packet

Dependent packets

Use "sudo apt install" to install "build-essential、libqt4-qt3support、 itcl3、itk3、iwidgets4" one by one.

99-coolsand-dongle.rules file

Copy 99-coolsand-dongle.rules file into path of "/etc/udev/rules.d/".

3.1.2.2 COM Configuration

Create a new folder named "comport" in the folder which coolwatcher_*.exe is located.

Create symbolic link file for /dev/ttyUSB0, /dev/ttyUSB1 and /dev/ttyUSB* in comport folder. As following:

ln –s /dev/ttyUSB0 comport/COM1

ln –s /dev/ttyUSB1 comport/COM2

Notice that COM1, COM2 should be capital letter. And the COM number should be in COM1 to COM255.

3.2 USB Mode

Now the USB mode only supported in 8910 module.

3.2.1 Windows

Install the USB driver in DTKWin\drivers\8910usb folder, the 8910 module is suggested to

connect with PC before install driver. After USB driver installed, 8 USB serial ports can be checked in device manager, and the Port4 is used to communicate for coolwatcher.

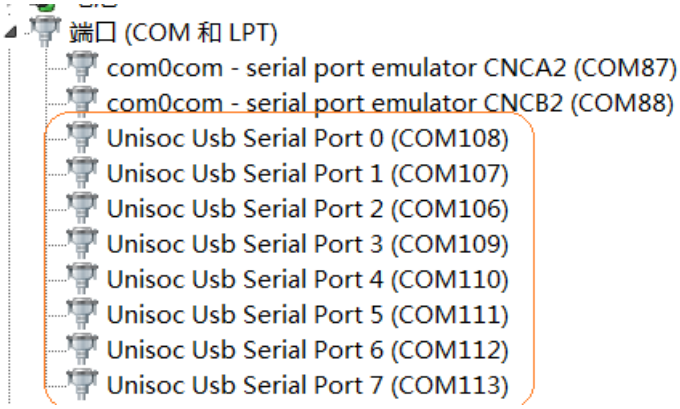


Figure 3-2-1 8910 USB ports

3.2.2 Ubuntu16.04

3.2.2.1 Driver Configuration

Default driver can be used.

The USB serial port has no standard CLASS, PID/VID is used to recognize the USB under Linux.

- Configuration in cmd line

  a) Mount option driver

  /sbin/modprobe option

  b) Add the modes, may be used in 8910 module, to option recognized list

  echo 1782 4d10 ff > /sys/bus/usb-serial/drivers/option/new_id

  echo 1782 4d11 ff > /sys/bus/usb-serial/drivers/option/new_id

- Modify Kernel code

  File to be modified is drivers/usb/serial/option.c, add followings in option_ids:

```
// Eight Serials
{ USB_DEVICE_AND_INTERFACE_INFO(0x1782, 0x4d10, 0xff, 0, 0) },
// netdev and four serials
{ USB_DEVICE_AND_INTERFACE_INFO(0x1782, 0x4d11, 0xff, 0, 0) },
// diag device
{ USB_DEVICE_AND_INTERFACE_INFO(0x1782, 0x4d11, 0xff, 0, 0) }
```

3.2.2.2 Port

Identify port

ttyUSB{*} device can be obtained int /dev/ folder after USB plunged，such as ttyUSB0、ttyUSB1、…、ttyUSB7. tty device do not display its device name under Linux, so, the udevinfo should be used to confirm device number.

```
udevadm info –n /dev/ttyUSB{0-7}
```

Checking ID_USB_INTERFACE_NUM=${id}. If the pid is 4d10, Ids should be 0-7, the communication port between 8910DM and coolwatcher is generally ttyUSB4. If pid is 4d11, id should be 2-9, the communication port between 8910DM and coolwatcher is generally ttyUSB6. The ports are corresponding to *Port0 - *Port7 under Windows.

Adaptation coolwatcher

Create a new folder named “comport” in the folder which coolwatcher_*.exe is located.

Create symbolic link file for /dev/ttyUSB0, /dev/ttyUSB1 and /dev/ttyUSB* in comport folder. As following:

```
ln –s /dev/ttyUSB4  comport/COM5   // pid 4d10
```

ln –s /dev/ttyUSB6 comport/COM7 // pid 4d11

Notice that COM6, COM7 should be capital letter. And the COM number should be in COM1 to COM255.

3.3 Startup Coolwatcher

coolwatcher_debughost.exe/ coolwatcher_usb.exe are consistent in startup and functions.

Double-click the <coolwatcher.exe>, and you can enter the start configuration interface, Figure 3-3-1. Profiles at left are phone/module type, and right are configuration options.

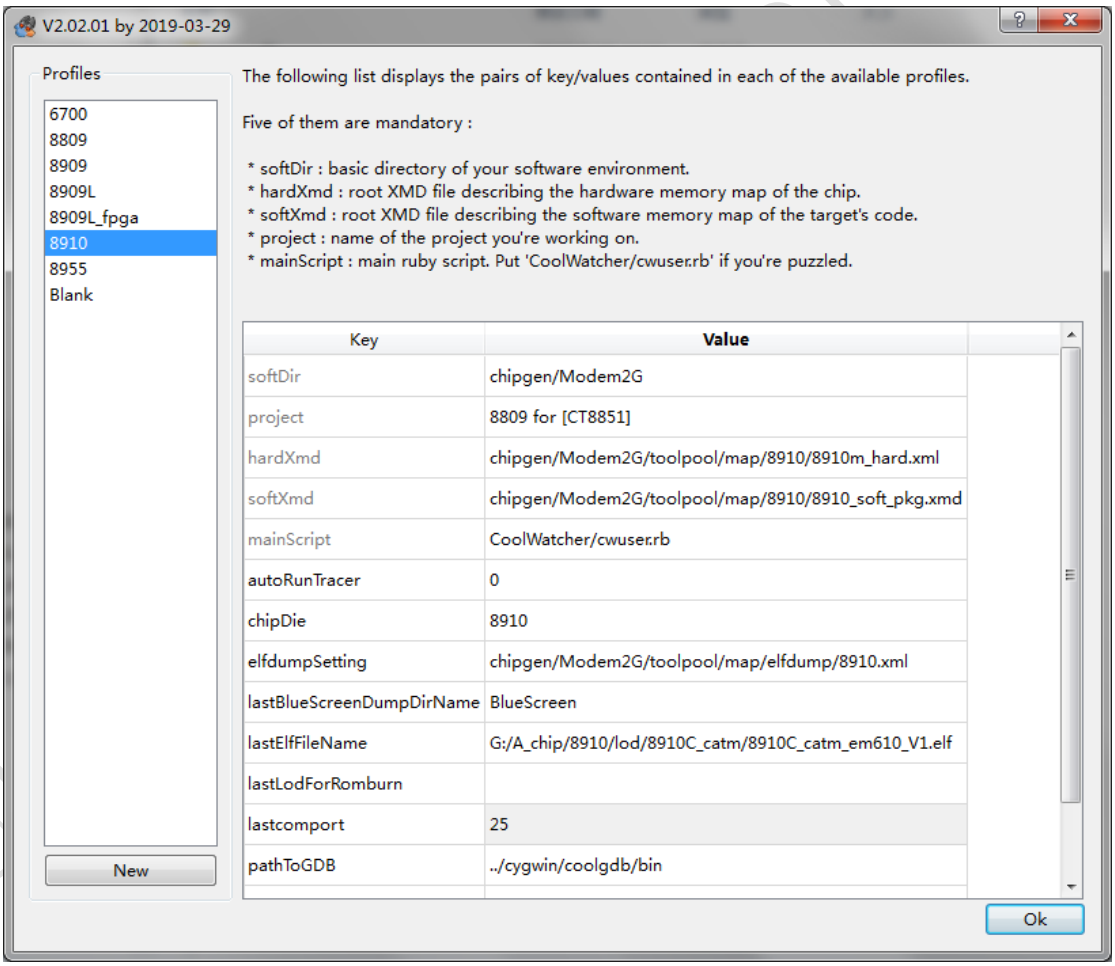


Figure 3-3-1 Coolwatcher Setting interface

1. Choose the type of phone/module, e.g.: UIS8910DM choose the 8910.
2. Modify the related configuration options, e.g.: port number.

Click OK, then "coolhost" interface will be opened, shown as Figure 3-3-2.

3. Coolhost setting

COM port and band rate already set in interface 3-3-1, now only the Flow Control should be set.

Flow Control should be XON/XOFF under debughost mode, and cannot be modified, shown in Figure3-3-3.

Flow Control should be NONE under USB mode, and cannot be modified, shown in Figure3-3-4.

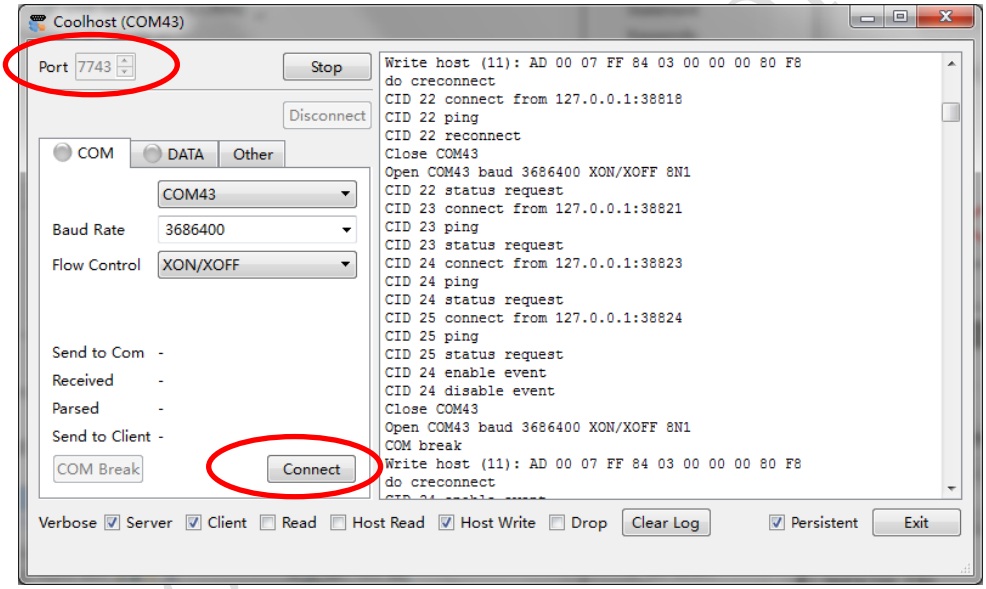


Figure 3-3-2 coolhost interface

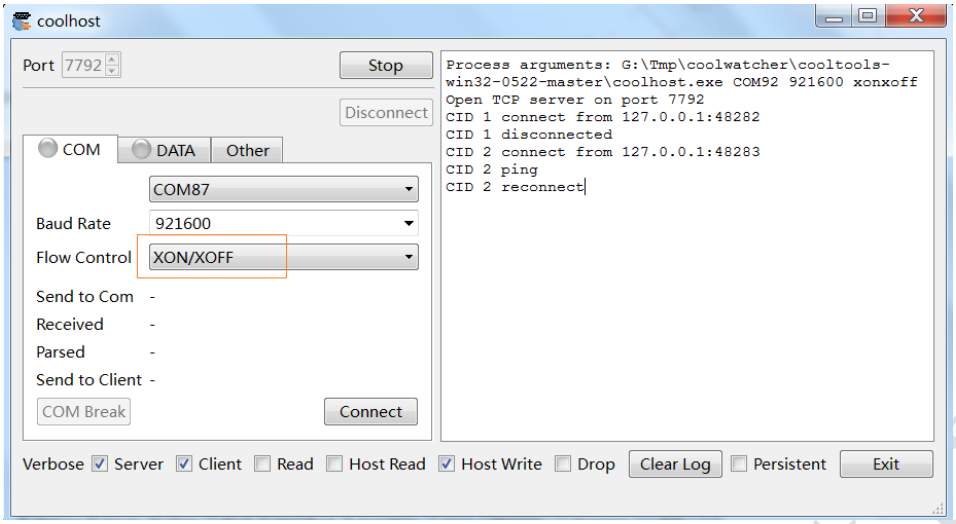


Figure 3-3-3 coolhost interface in debughost mode

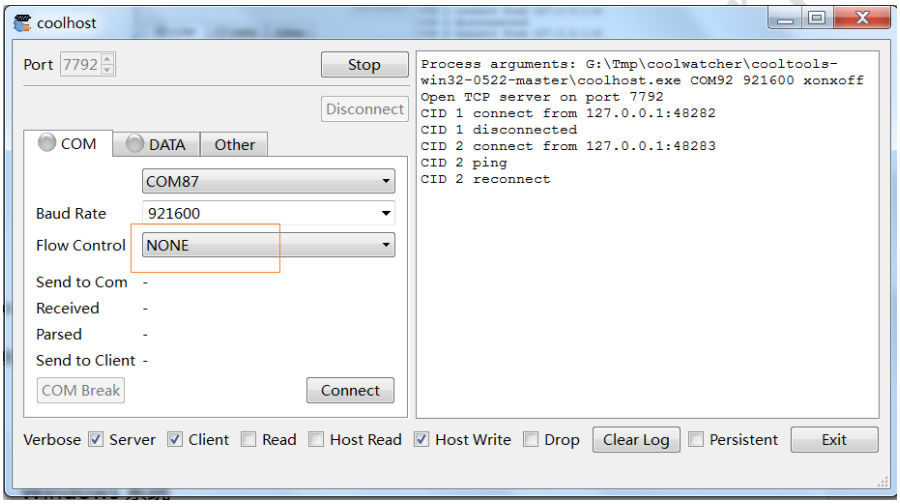


Figure 3-3-4 coolhost interface in USB mode

4. AT command

Debughost mode do not use to send any at command.

USB mode should send at command "AT^TRACECTRL=0,1,2" to activate USB log.

Port0 shown in Figure 3-2-1 is the AT port.

4 Function Instruction

The Coolwatcher main interface is shown as Figure 4-1.

Menu bar, Tool bar, Log print area, HW Library tree, SW Library tree, Register watcher,

Buffer watcher and information area are displayed.

Coolwatcher can burn flash, watch buffer information, read register, view trace and dump memory data.

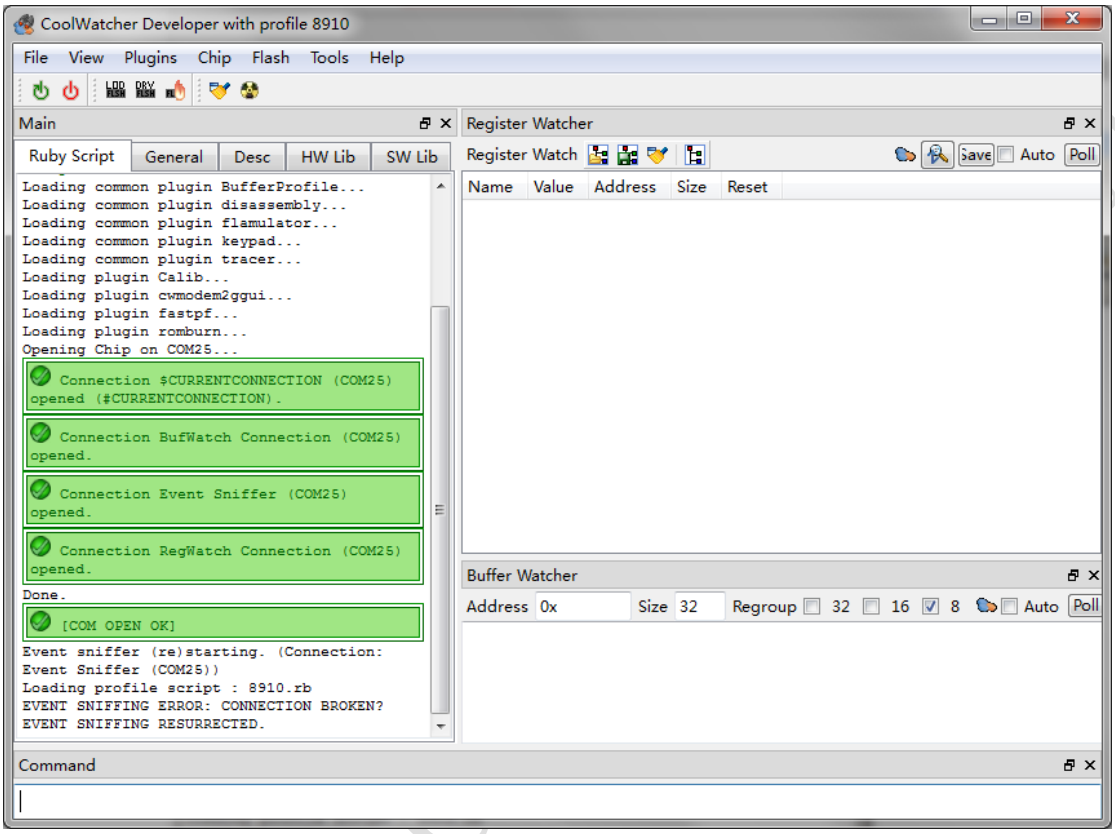


Figure 4-1 coolwatcher main interface

4.1 Burn Flash

Note: This function do not supported on 8910 and 8910L. 8910 and 8910L burning flash can use ResearchDownload or FACTORYDOWNLOAD.

4.2 View Buffer Information

Input data in “Address” and “Size” on “Buffer Watcher area”, then click <poll> can get related data from cellphone/module.

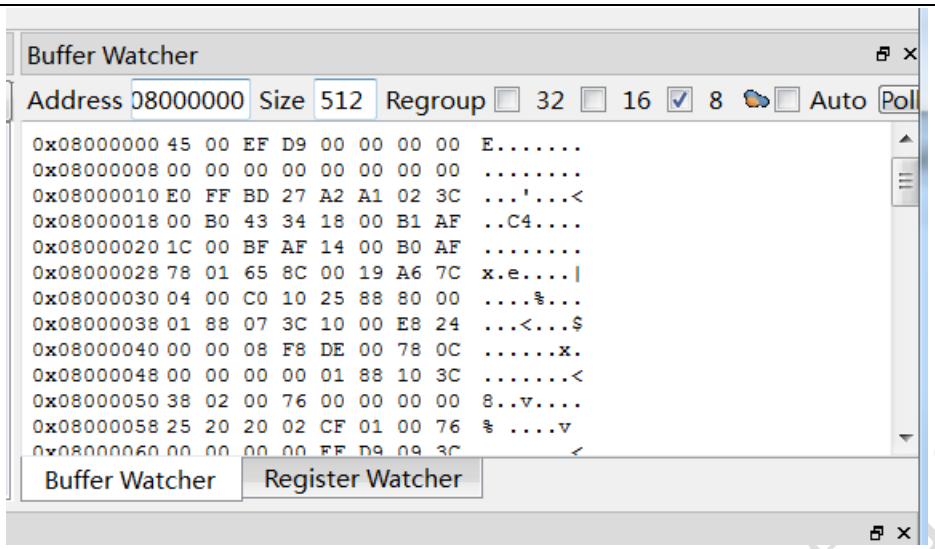


Figure 4-2-1 Buffer watcher interface

4.3 View Register

Drag registers from "HW Library" or "SW Library" to "RegisterWatcher area", then click <poll> can get related data.

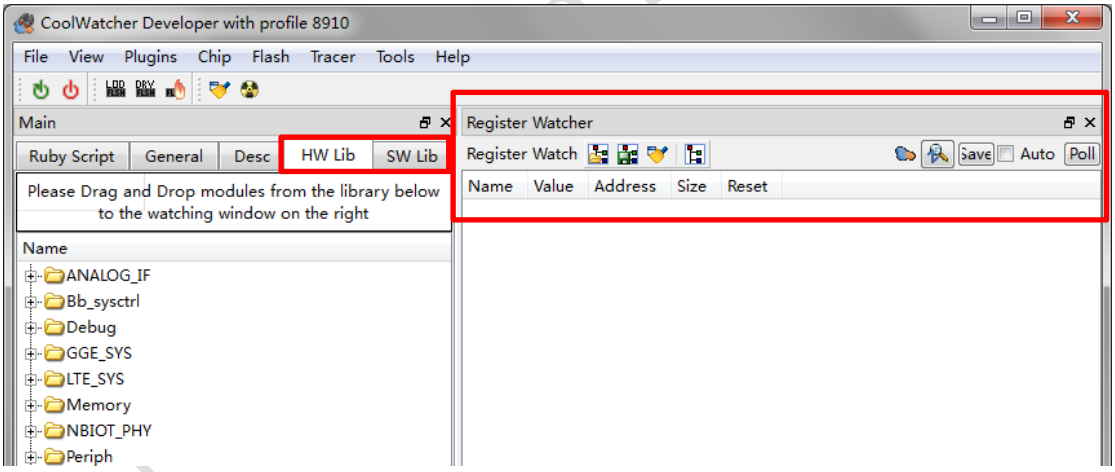


Figure 4-3-1 Register View Interface

4.4 View Trace

4.4.1 Activate Tracer

Click <Plugins> —> <Activate Tracer> on menu bar, and activate Trace, shown as Figure

4-4-1. And the tracer main interface is shown as Figure 4-4-2.

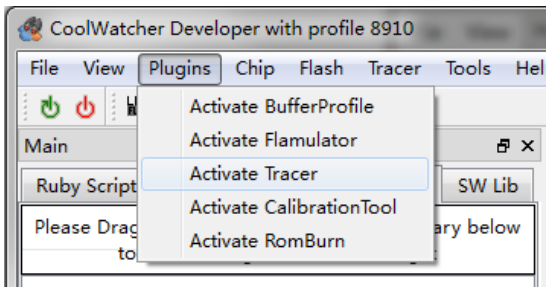


Figure 4-4-1 activate tracer

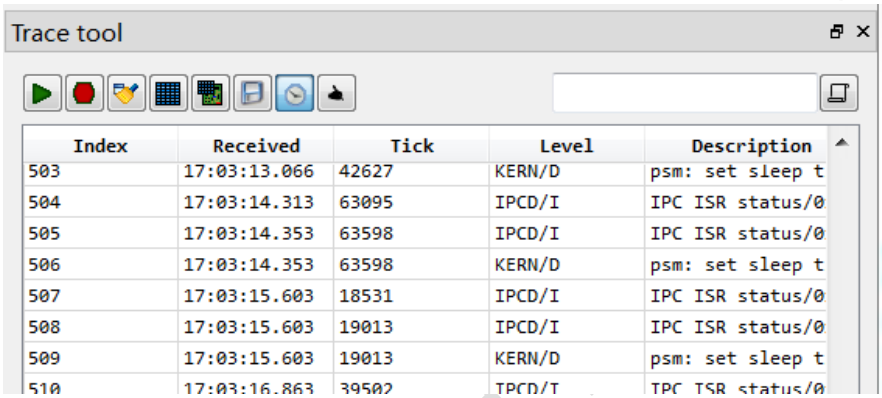


Figure 4-4-2 Tracer main interface

Tool bar can help you start follow actions quickly: Start Trace, Stop Trace, Clean Trace, Set Trace Levels, Reapply trace levels, Save Trace, Start/Close Received Row, Start/Close Comment, and so on.

In the middle of the interface, Trace Index, Time that Trace Received by PC, Levels, and Descriptions can be obtained.

4.4.2 Tracer Menu

When the Tracer Tool is activated, the following interface is displayed.

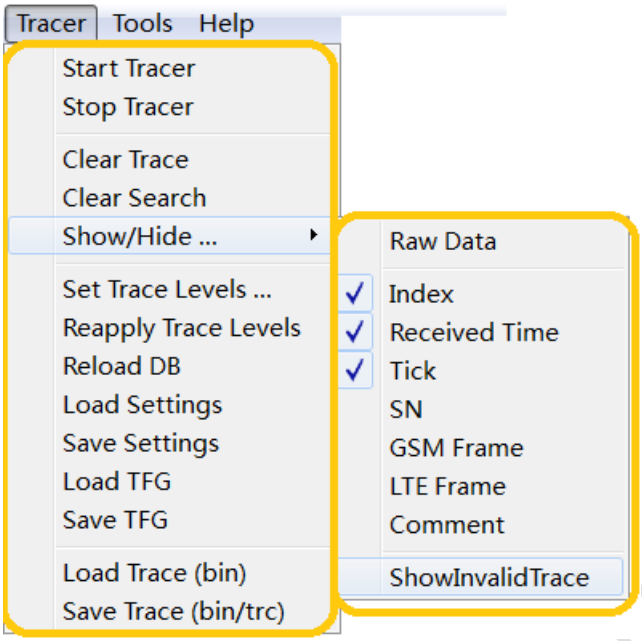


Figure 4-4-3 tracer menu items

Menu Items:

LoadSettings：load Levels setting file;

SaveSettings：save Levels setting file;

Load TFG：load tfg file;

Save TFG：save tfg file;

Load bin: load hex trace file and replay it;

Save Trace（bin/trc)：save trace to bin/trc file.

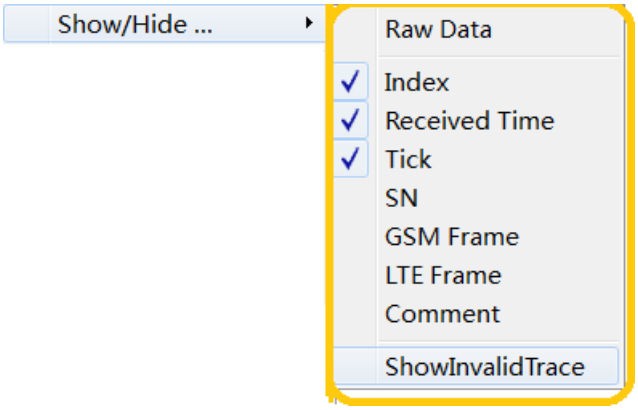


Figure 4-4-4 Tracer Show/Hide menu

When a new item is selected in Figure 4-4-4, a new row is displayed in Figure 4-4-2.

Toggle Raw Data: Start/stop Raw data, and shown as Figure 4-4-5.

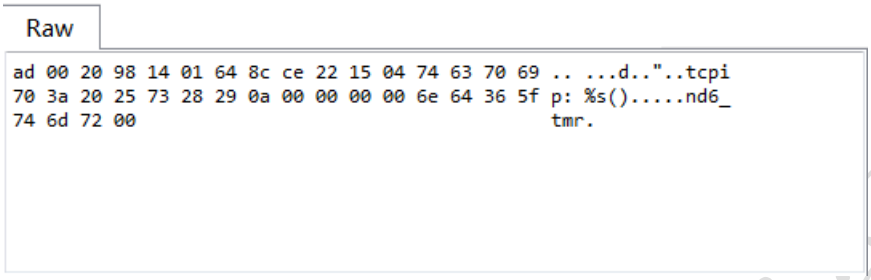


Figure 4-4-5 Raw Data interface

SN: SN is a 16bits number, can be used to judge trace lose or not. The SN row 是 is empty, when trace do not loosed.

ShowInvalidTrace: start / stop invalided trace. It contains UnusedTraceID, ParseFailed, and SeqJump three items.

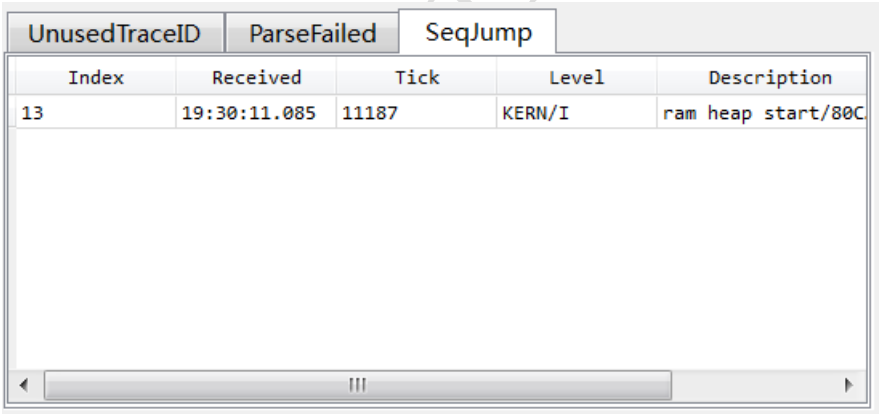


Figure 4-4-6 invalided trace interface

4.4.3 Instruction for Tracer

1. Trace Levels Setting

Click the forth button  , which located at the trace main interface, set Trace Levels. Figure 4-4-7 shows the setting interface.

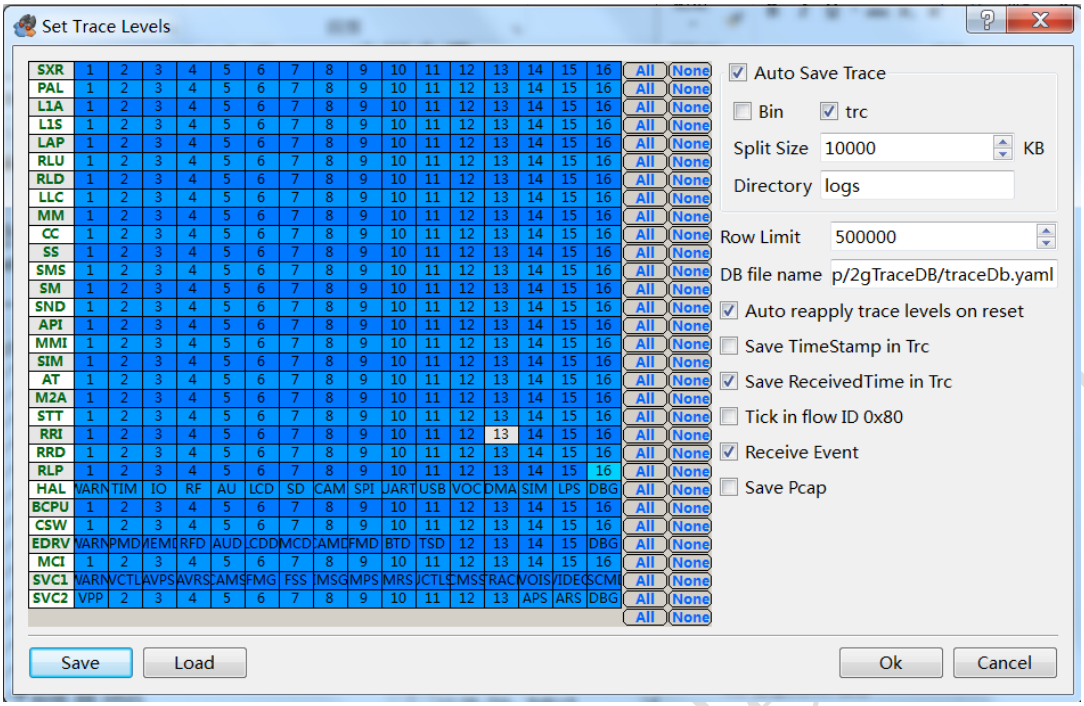


Figure 4-4-7 trace setting interface

AutoSave

- Check trace is auto saved or not, checked means auto saved.
- bin.trc: type of trace file, bin means binary file, and trc mean text file.
- Split Size: size of trace file. When file size is larger than this value, file will be automatic segmentation.

DB file name

That is trace id DB file.

RowLimit

Max line number in tracer tool main interface.

Auto reapply trace levels on reset

Auto reapply trace levels when reset chip.

Save TimeStamp in Trc

Save timestamp in *.trc trace file.

Save ReceivedTime in Trc

Save timestamp in *.trc trace file.

Tick in flow ID 0x80

This configuration should keep in touch with lod, select this item when the lod has timestamp.

ReceiveEvent

Save Pcap

Save pacp traces into *.pcap file.

Buttons at left bottom corner:

Save：Save Levels config.

Load：Load Levels config.

2. Click the first button  in Figure 4-4-8, and start Trace tool. Trace information will displayed in table.

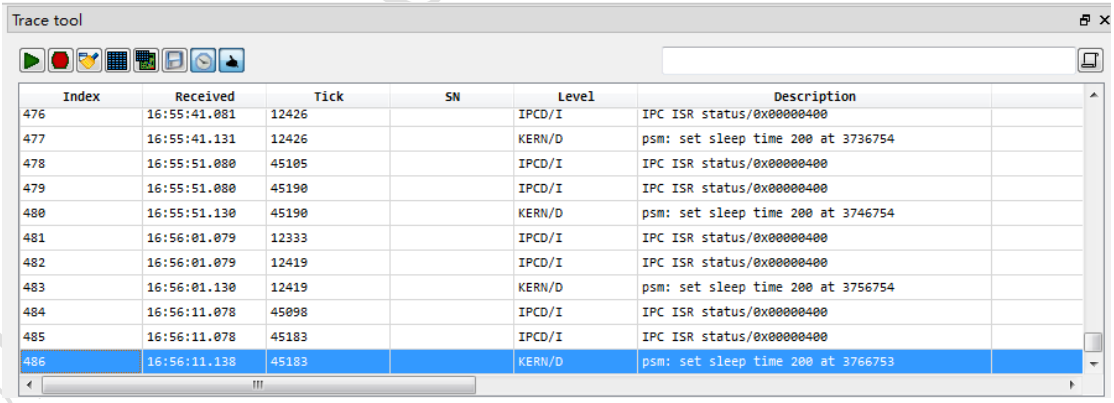


Figure 4-4-8 tracer view interface

3. Click the second button  in Figure 4-4-8, Trace tool can be stopped.

Note:

Configuration of Tracer is auto saved in “rbbase/common/plugins/tracer/” folder.

4.5 GDB Launcher

GDB is an important method in phone hang and trace problem analysis.

4.5.1 Activate GDB

Click <Tools> —> <GDB Launcher> at menu bar, and start the following configuration frame.

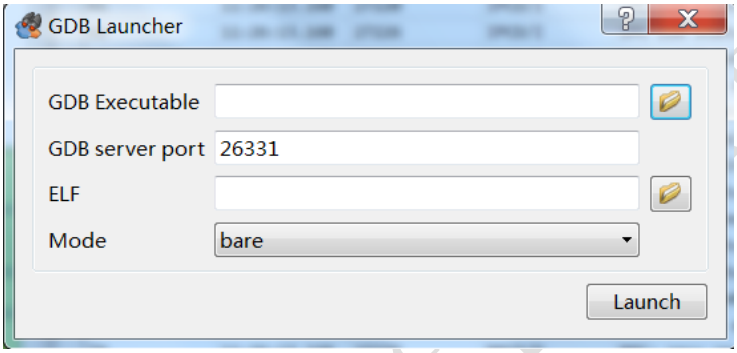


Figure 4-5-1 GDB interface

- GDB server port: default value, do not to modify;
- ELF file: select the elf according to chip version;

GDB Executable: related to Mode, when select mode as 8910AP or 8910CP, GDB Executable cannot be set. When select other mode, select GDB Executable as mips-elf-insight.exe in mips-gdb/bin folder.

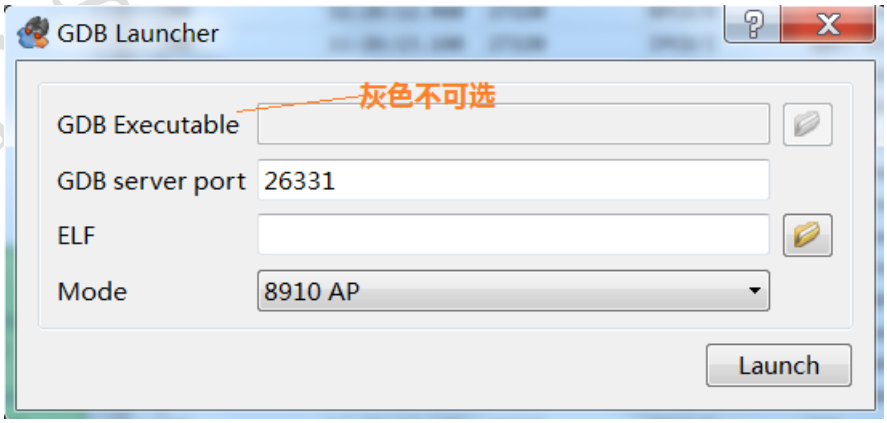


Figure 4-5-2 GDB Executable cannot be selected

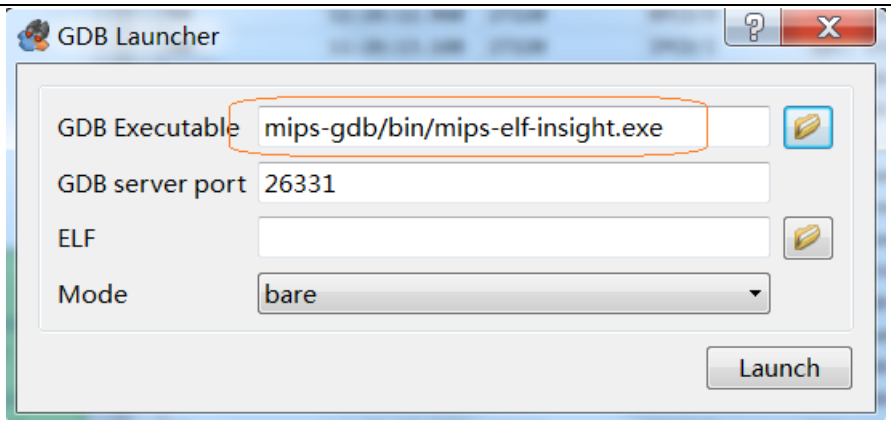


Figure 4-5-3 GDB Executable is selectable

- Mode:

  There are seven modes: "bare", "sx(with REDUCED_REGS)", "sx(without REDUCED_REGS)", xcpu_rom, live, 8910 AP, and 8910 CP. When "sx" mode is compared with "bare" mode, "thread info" and "thread" this two command are added for "sx" mode. "xcpu_rom" mode is always used when analyze the hang that accrued before system booting. "live" mode only used when analyze serious cellphone hang problem.

  - UIS8910DM can only use 8910 AP or 8910 CP mode.
  - Other module select "bare" mode in most condition.

Click the <Launch> button in Figure 4-5-1, the GDB code interface will be loaded as Figure 4-5-4.

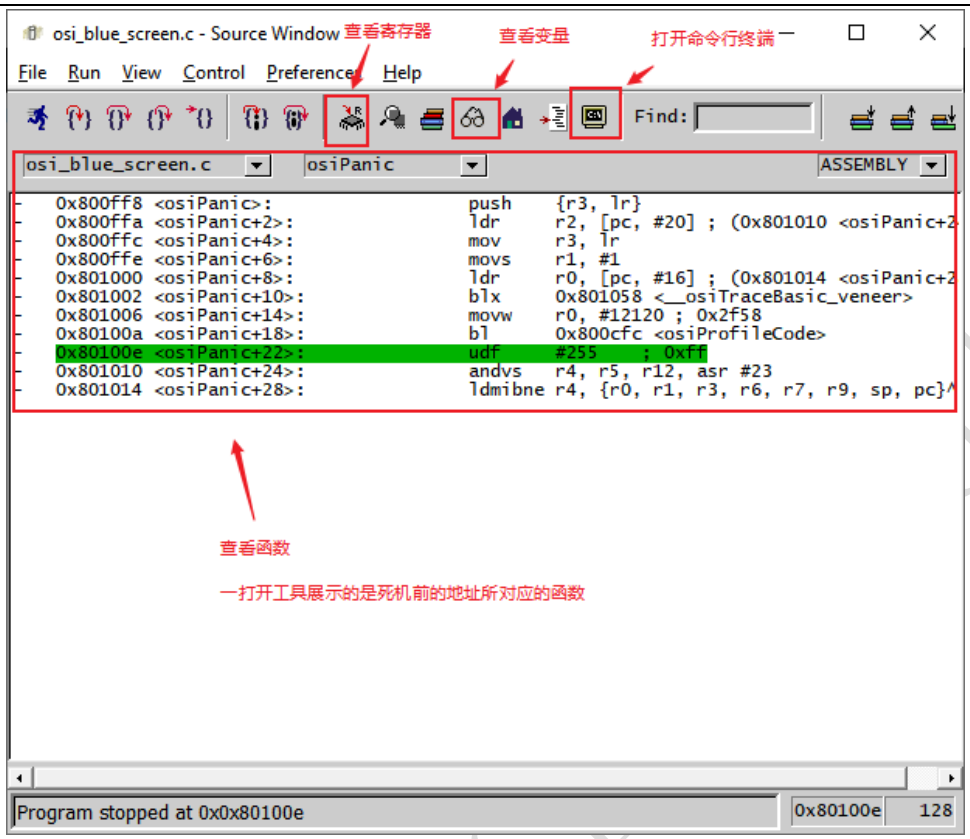


Figure 4-5-4 GDB code interface

4.5.2 Get Stack Information

Click the <Console> button, the 14th button  in Figure 4-5-4 tool bar, and open the gdb command window.

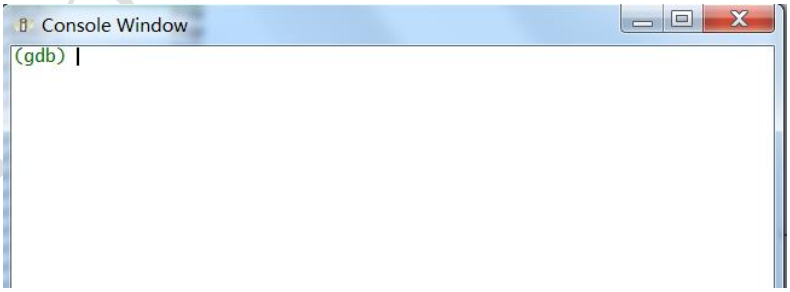


Figure 4-5-5 GDB command interface

Open the GDB console window, and input the gdb command, then the specific information will be displayed. The most used is “bt”, it can be used to check back trace. “bt f” can display more detail informations.

Figure 4-5-6 GDB analysis result

4.5.3 Check Variable

Click the <Watch Expressions> button, the 11th button  in Figure 4-5-4 tool bar, and open the watch window. The global symbol can be checked. If the address of the symbol is known, it can be cast to specific data types.

Figure 4-5-7 GDB watch window

4.5.4 Check Registers

Click the <Registers> button, the 8th button  in Figure 4-5-4 tool bar, and open the registers window. All CPU registers can be checked in this window. In 8910 module, ARM registers are displayed.

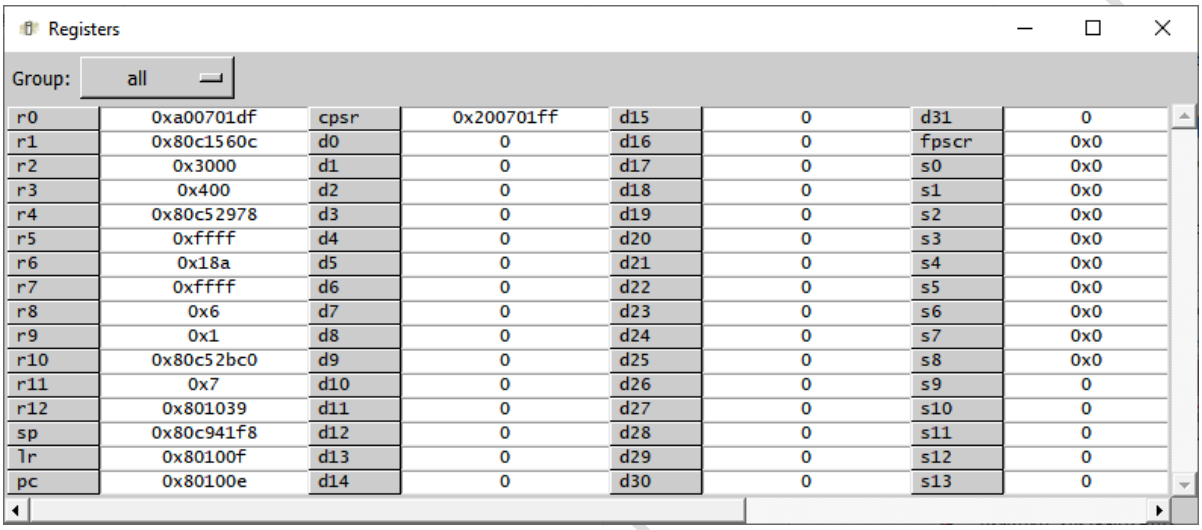


Figure 4-5-8 GDB registers window

4.5.5 GDB Common Command

ID	GDB command	Note
1	p<sth>(print <sth>)	print the value of sth, sth can be an expression, variable, pointer, and so on.
2	display <sth>	the same as p<sth>, but display<sth> will display the value of sth.
3	bt or bt f (backtrace or backtrace full)	Aacktraces (backtraces full) the current executed code. You get the call stack, parameters passed to each function, & so

		on. By using "full" you will also get the display of all local variables for these functions (EXTREMELY useful).
4	up	Goes up into the call stack. To be used in conjonction with bt.
5	down	Goes down into the call stack. To be used in conjonction with bt.

4.6 Profile Analyze

Profile is an important method to analyze problems. Profile displays what the CPU doing when the software running. Such as which task is working or which interrupt is working in certain tick.

Different use instrument fit for different module.

4.6.1 Save *.prf Files

4.6.1.1 8910 and Following Modules

Click Tools -> Profile Dump (v2.0) to start Profile Dump, shown as Figure 4-6-1.

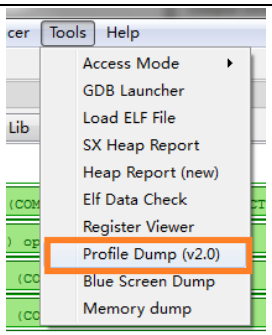


Figure 4-6-1 start Profile Dump (v2.0)

Choose the proper ELF file and output file name, then click <OK>.

Figure 4-6-2 Profile Dump main interface

When following information displayed represents the dump is successful.

```
Find gProfileCtx in elf 0x80c2160c
Profile buffer: 0x80c1560c
Profile size:   12288
Profile start:  1025
Profile count:  12288
Dump profile to F:/WORK/bugs/8910/1253880/11_2_pc3_com52_reboot_91h/a.prf
Dump profile finished
```

Figure 4-6-3 Profile Dump success tip

4.6.1.2 8955/8908/8909 .etc.

Click the <Plugins> -> <Activate BufferProfile> in coolwatcher main interface tool bar to activate buffer profile plugin, shown as Figure 4-6-4. The BufferProfile menu will list in menu bar, and relevant button will added to the tool bar.

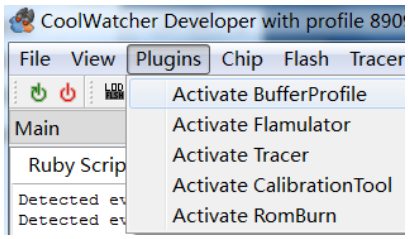


Figure 4-6-4 activate buffer profile

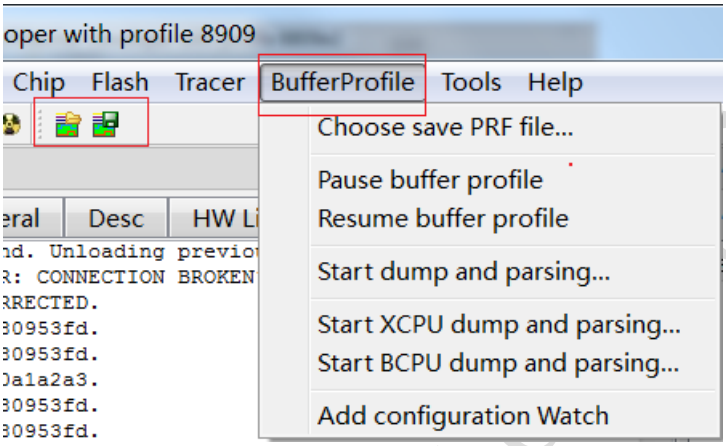


Figure 4-6-5 bufferProfile button on tool bar and menu bar

When the mobile/module shutdown, the buffer profile plugin can be used to get relevant information. The method is as follows:

1. Phone connected with PC by COM port, and make sure communication is ok.

2. Click <Choose save PRF file> menu, or click the button on toolbar, enter the *.prj file. The default file path is the same as coolwather.exe.

3. Click <Start XCPU dump and parsing> menu, begin to save information about XCPU. The information saved as follows.

```
> parseProfilerGo(PARSE_ALL_PROFILE)
XCPU Buffer Address = 0x824747cc
XCPU Record Position = 0x22ed
XCPU Record Number = 0x3000
Dump XCPU profile ...
Dumping ...
Done.

0.094000 0.078000 0.172000 ( 2.919000)

Parse XCPU profile ...
1 - File Size = 49243
1 - Buffer Size = 49152
1 - Buffer Position = 35764
1 - Buffer freq = 16384
Begin parsing memory profile ...
count = 49152
Parse done: F:/cooltools-win32/test_r630.prf
```

Figure 4-6-6 save XCPU information

4. Click <Start BCPU dump and parsing> menu, begin to save information about BCPU. The information saved as follows.

```
BCPU buffer profile exists!
BCPU Buffer Address = 0xa1983ab0
BCPU Record Position = 0x32
BCPU Record Number = 0x40
Dump BCPU profile ...
Dumping ...
Done.

0.000000 0.015000 0.015000 ( 0.037000)

Parse BCPU profile ...
1 - File Size = 343
1 - Buffer Size = 256
1 - Buffer Position = 200
1 - Buffer freq = 16384
Begin parsing memory profile ...
count = 256
Parse done: F:/cooltools-win32/test_r630_bcpu.prf
```

Figure 4-6-7 save BCPU information

5. Click <Start dump and parsing> menu, will save both BCPU and XCPU. The information saved as follows.

```
Combine and parse both XCPU and BCPU profile ...
1 - File Size = 49243
1 - Buffer Size = 49152
1 - Buffer Position = 35764
1 - Buffer freq = 16384
2 - File Size = 343
2 - Buffer Size = 256
2 - Buffer Position = 200
Begin parsing memory profile ...
count = 49408
Parse done: F:/cooltools-win32/test_r630_all.prf
```

Figure 4-6-8 save all information

Note:

If phone is not crashed, it is better to click <Pause buffer profile> before capture *.prj file. After file is saved, click <Resume buffer profile> to resume.

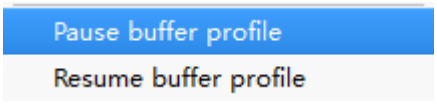


Figure 4-6-9 save profile files without crash

4.6.2 Check Profile Information

Open coolprofile.exe tool. The main interface shown as Figure 4-6-10.

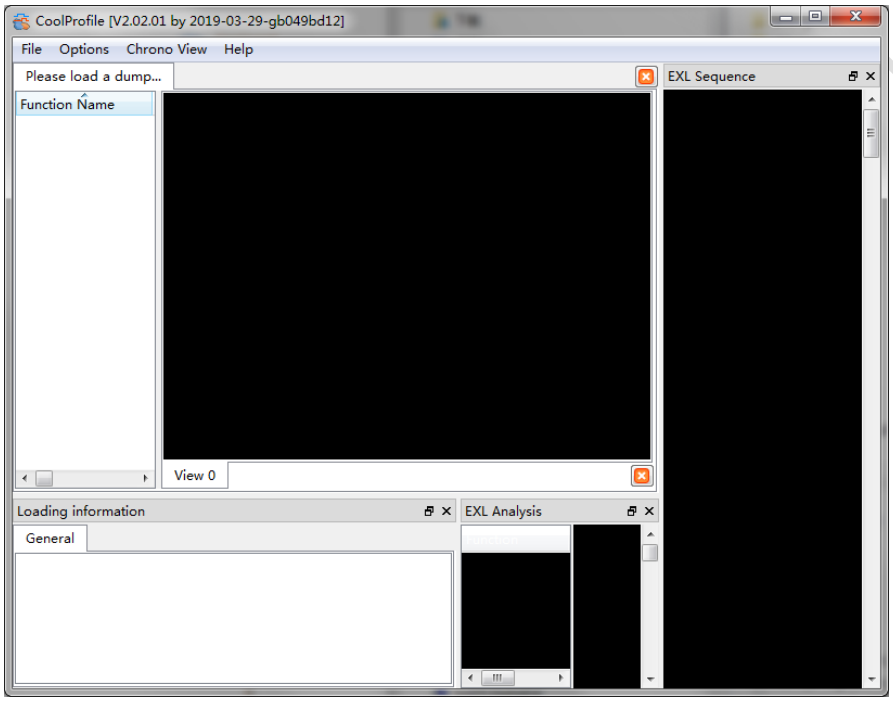


Figure 4-6-10 coolprofile interface

Open dump。Click <File> —> <Open dump> menu, the coolprofile dialog will opened, then choose dump and configuration files.

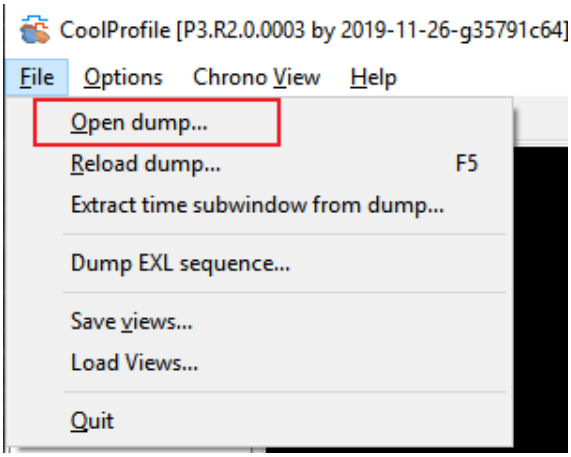


Figure 4-6-11 select open dump

Choose a configuration file: 8910 → 8910.xcp

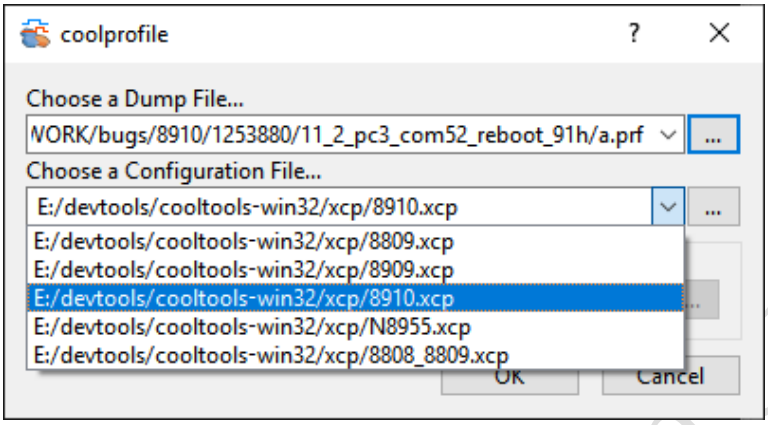


Figure 4-6-12 select profile files

Following Figure 4-6-13 will be displayed.

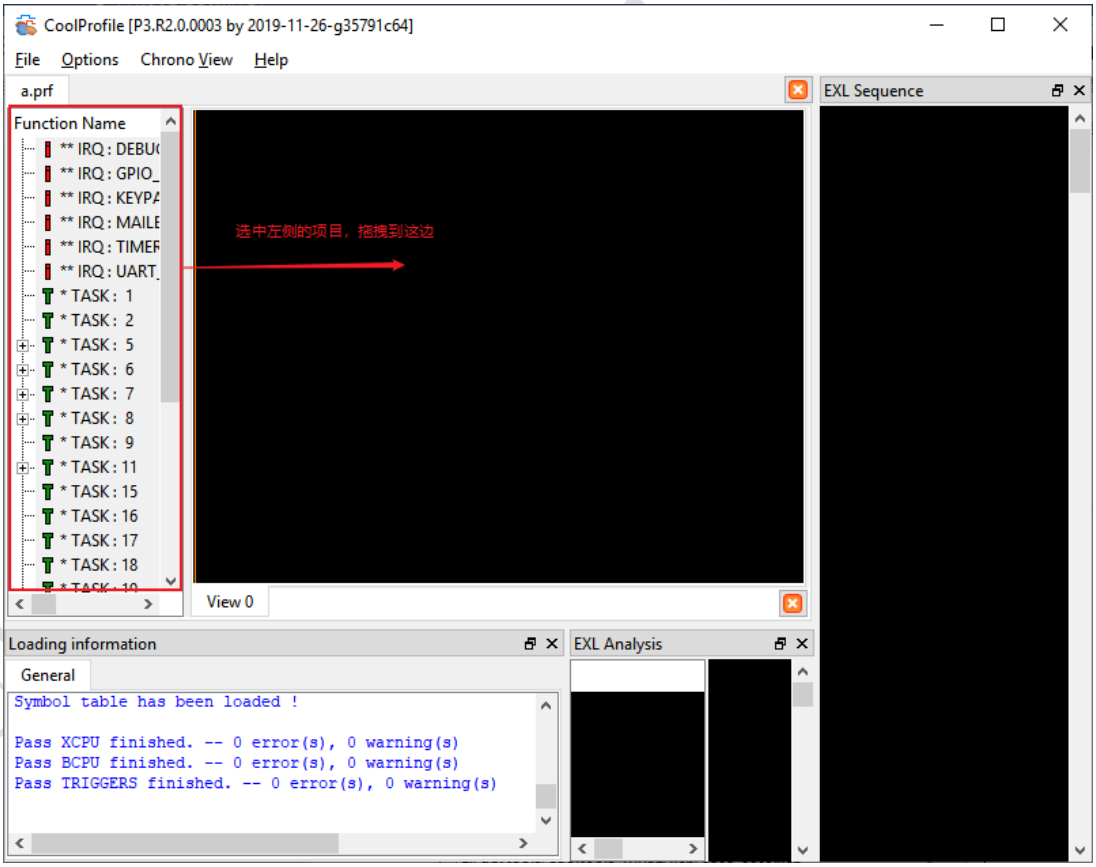


Figure 4-6-13 interface after load profile

Follow the prompts in Figure 4-6-13, drag functions to display area, then Figure 4-

6-14 will be displayed.

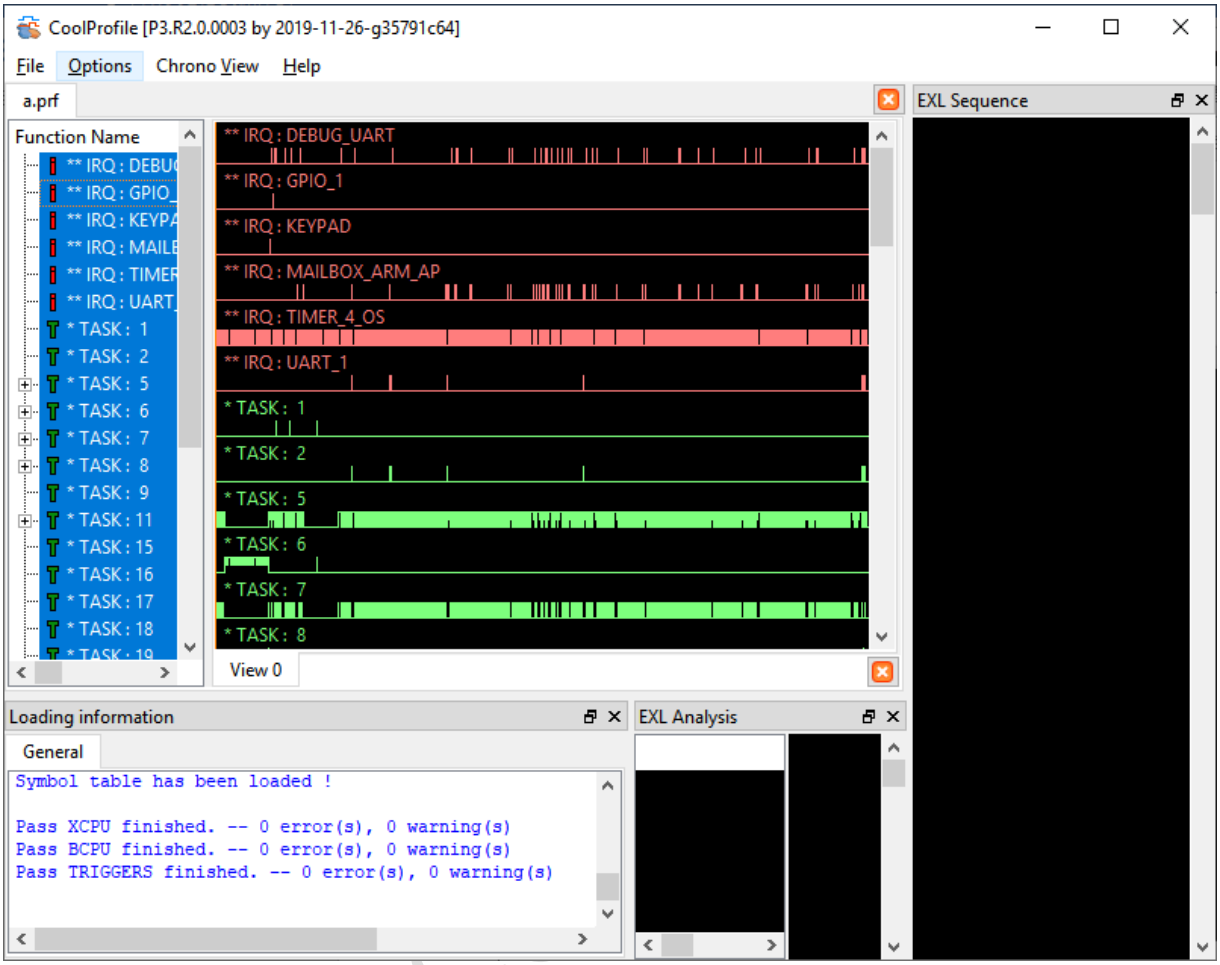


Figure 4-6-14 interface after loading files

Drag the mouse right button to enlarge the graph, and click the right button to shrink. Click the mouse left button can set a vertical line, and this line will disappear when click another place. But click the middle mouse button can set a fixed vertical line, which will not disappear when click the left button in another place. The high area of graph is the time of CPU running. Two vertical line can be used to calculate the certain time of code running.

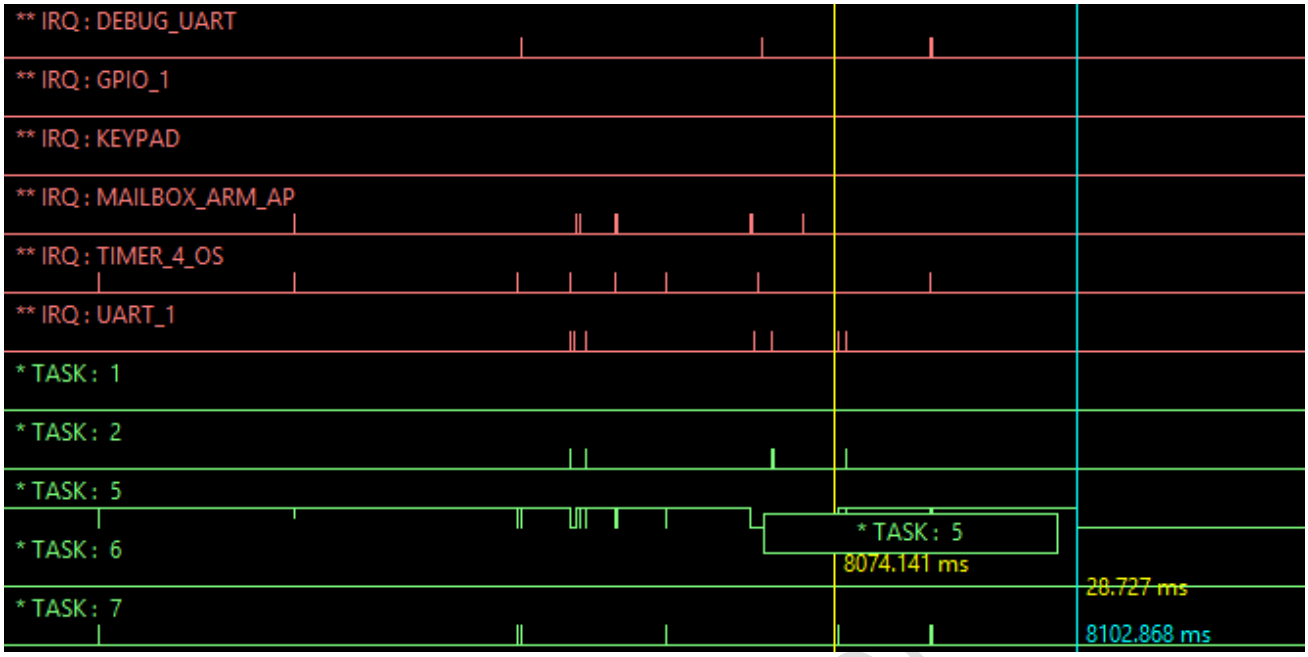


Figure 4-6-15 magnified diagram

4.7 Offline Analyze

4.7.1 Prepare for *.cmm or *.elf File

Refer to chapter 4.10.1 blue screen dump.

4.7.1.1 8910 and Following Modules

Refer to chapter 4.10 blue screen dump to save *.cmm file.

4.7.1.2 8955/8908/8909 .etc

After the phone hang, we can use elfdump command to save *.core(*.elf). Detailed commands please refer to chapter 4.14.

```
> elfdump "r630.elf"
building elf for 8809e2...
Reading page reg @0x81a0c000...
done
Reading xcpu reg @0x81a2b000...
done
Reading bcpu reg @0x8190a000...
done
```

Figure 4-7-1 Generate ELF file

4.7.2 Build Offline Analysis Environment

Coolwatcher support blue screen dump file import, and offline analysis.

When use the coolwatcher to analyze blue screen offline, it is suggest to close all coolwatcher and coolhost before build offline analysis environment.

4.7.2.1 8910 and Following Modules

Startup coolhost.exe as following interface.

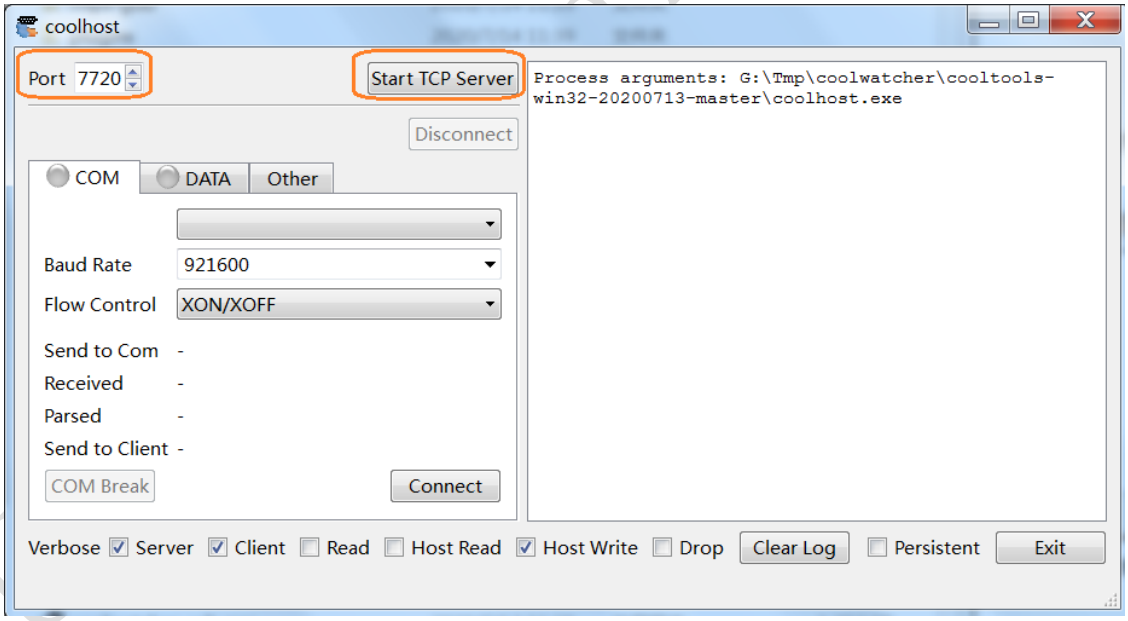


Figure 4-7-2 coolhost main interface

- **Port setting**

Port number can be set as any value, such as 20, the tcp port will be set as 7700 + COM number, shown in Figure 4-7-2 is 7720.

Then click Start TCP Server button.

- **Load blue screen dump file**

Open DATA page table, right click and select “Add”, as Figure 4-7-3, select *.cmm file, shown as Figure 4-7-4.

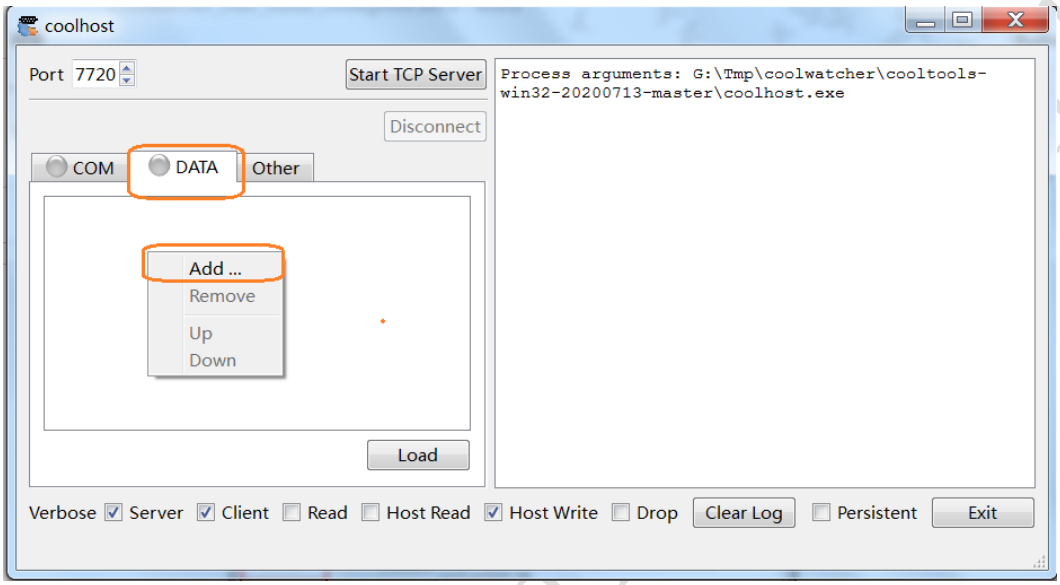


Figure 4-7-3 load dump file

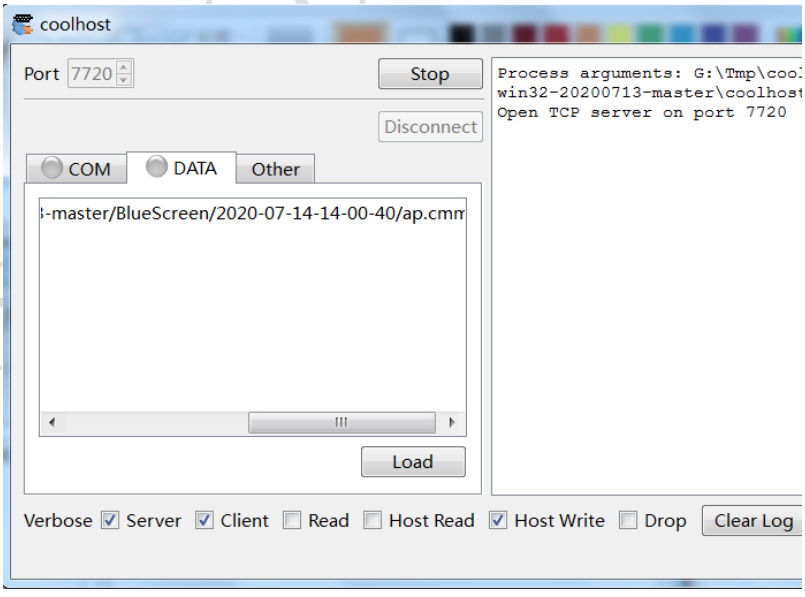


Figure 4-7-4 select *.cmm file

Click “Load” button in Figure 4-7-4 to load *.cmm file.

- **Startup coolwatcher**

Click coolwatcher*.exe, lastcomport set in following interface should be same with the port in Figure 4-7-4 Port 7720. If Port is set as 7720, the lastcomport must be 20.

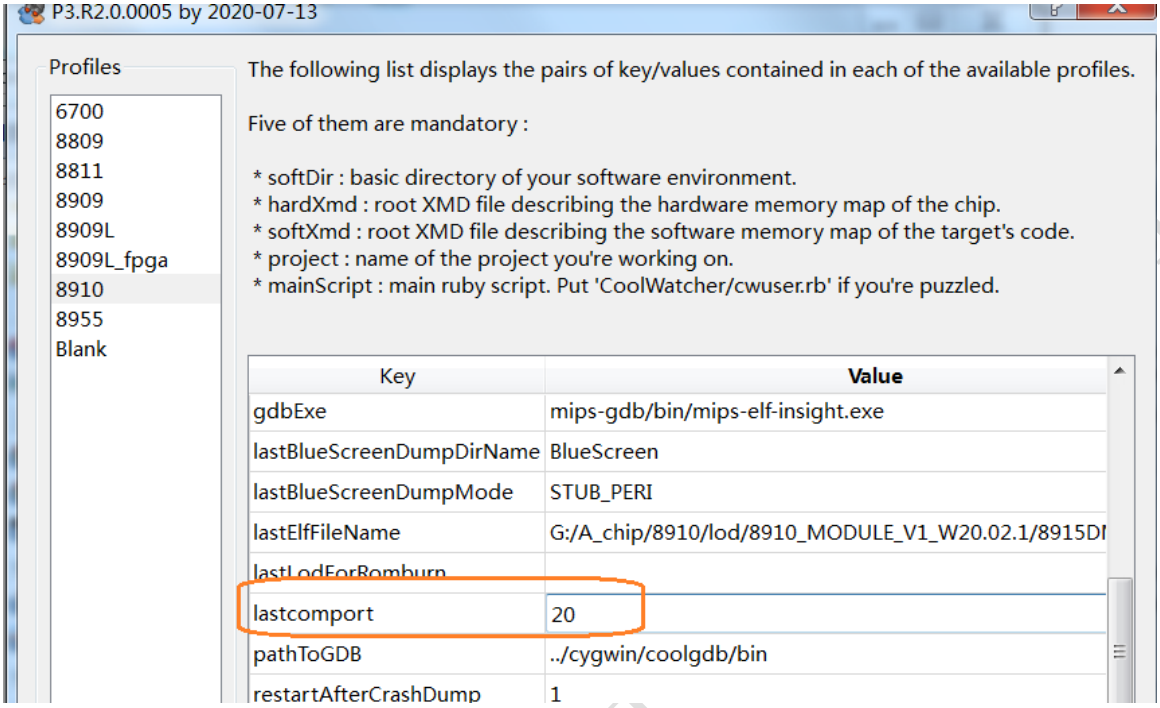


Figure 4-7-5 set lastcomport

Startup coolwatcher and analyze the offline problem。

4.7.2.2 8955/8908/8909 .etc

Startup coolhost.exe as Figure 4-7-2.

- **Port setting**

Port number can be set as any value.

Click Start TCP Server button.

- **Load blue screen dump file**

Open DATA page table, right click and select "Add", as Figure 4-7-6, select *.core (*.elf) and lod file, shown as Figure 4-7-7.

Notice that *.core (*.elf) should be select first. This *.core (*.elf) file is the file

that saved through elfdump command, which is not the elf file that generate with lod file.

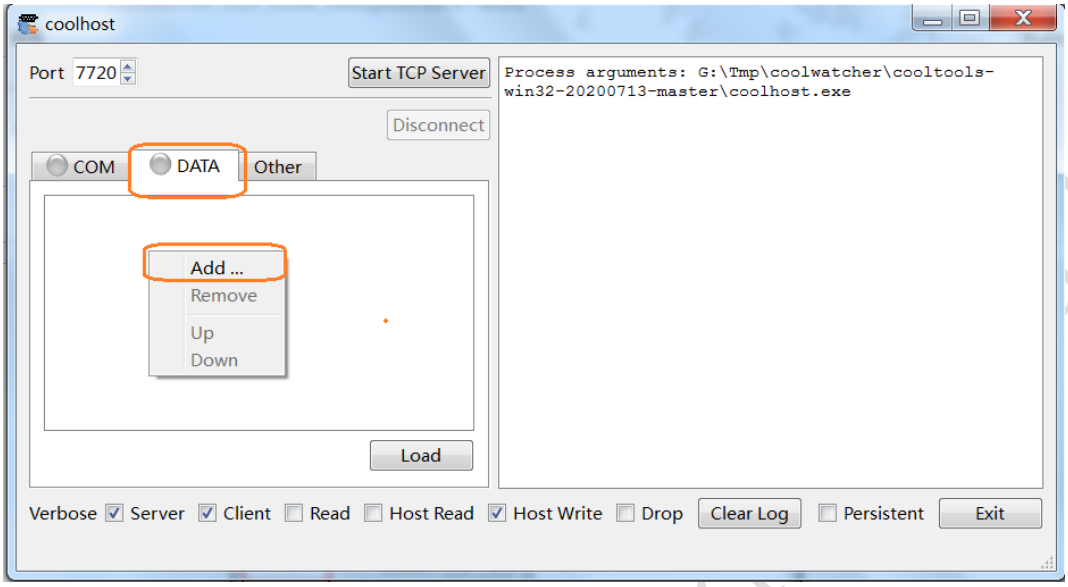


Figure 4-7-6 load *.elf file

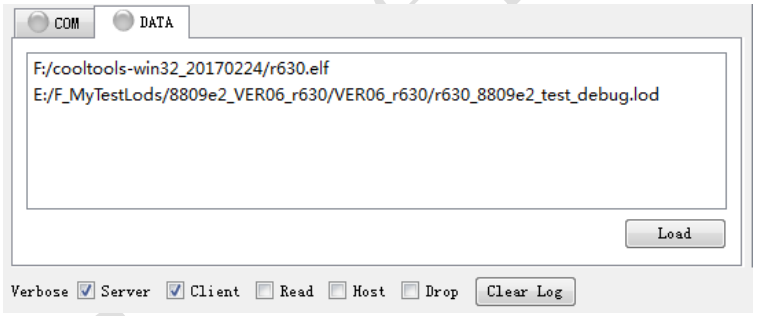


Figure 4-7-7 select *.elf file path

Click “Load” button in Figure 4-7-7 to load file, shown as Figure 4-7-8.

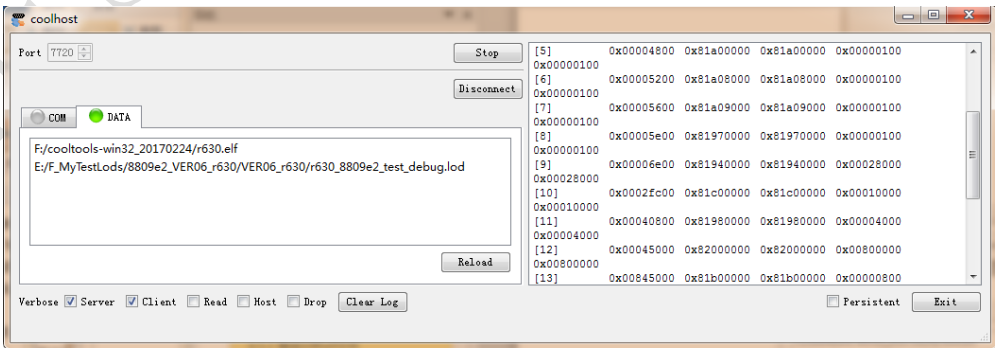


Figure 4-7-8 interface after load files

- **Startup coolwatcher**

Click coolwatcher*.exe, lastcomport set in following interface should be same with the port in Figure 4-7-6 Port 7720. If Port is set as 7720, the lastcomport must be 20.

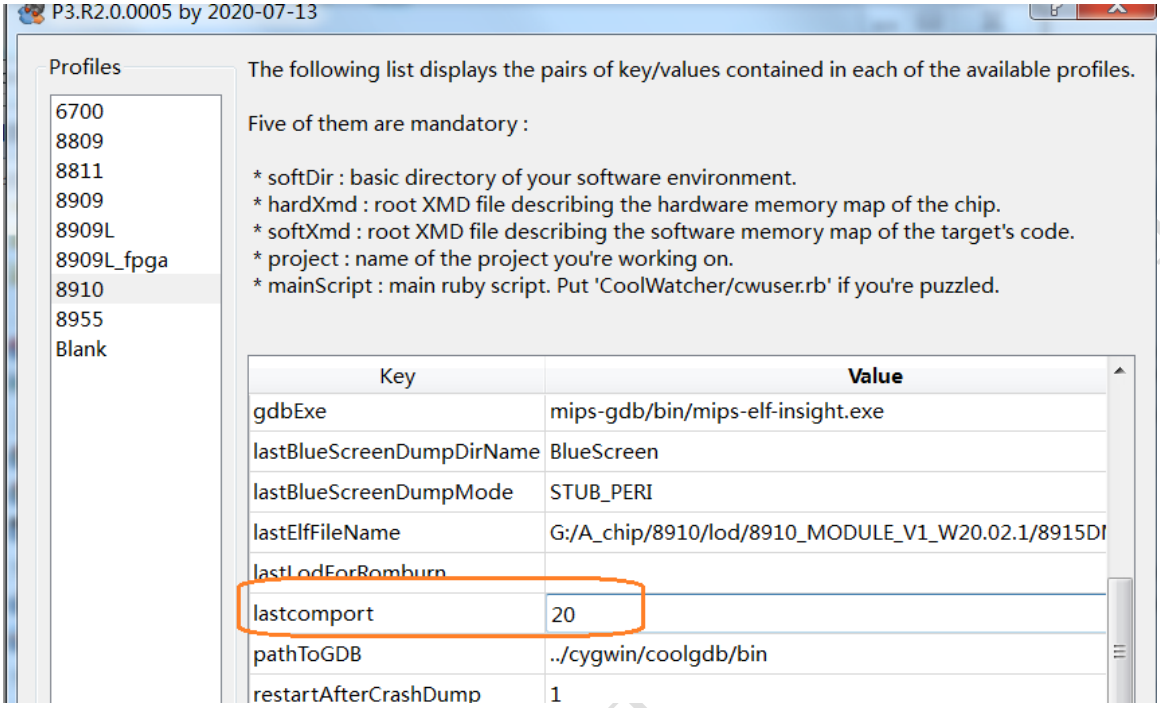


Figure 4-7-9 set lastcomport

Startup coolwatcher and analyze the offline problem。

4.7.3 Offline Analyze GDB

Keep the environment the same as 4.7.2, and refer to chapter 4.5 to analyze GDB.

4.7.4 Elf Data Check

This function is used in 8955/8908/8909 moudle.

Used to compare dump file and raw elf file, to check if code segment is overwrote.

Start

Set the environment the same as 4.7.2, click <Tools> —> <>Elf Data Check>, the UI shown as below, Figure 4-7-11.

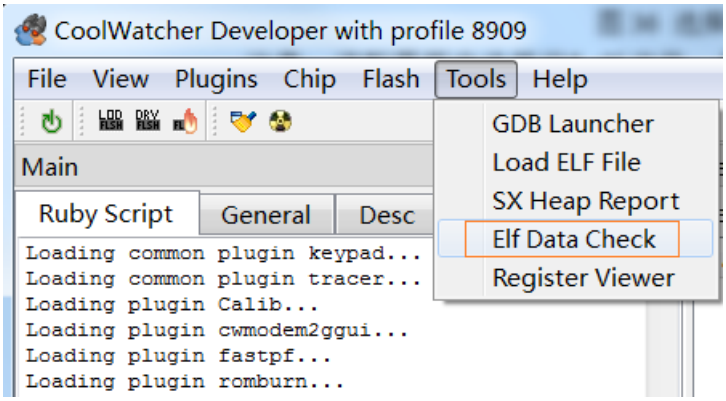


Figure 4-7-10 open Elf Data Check

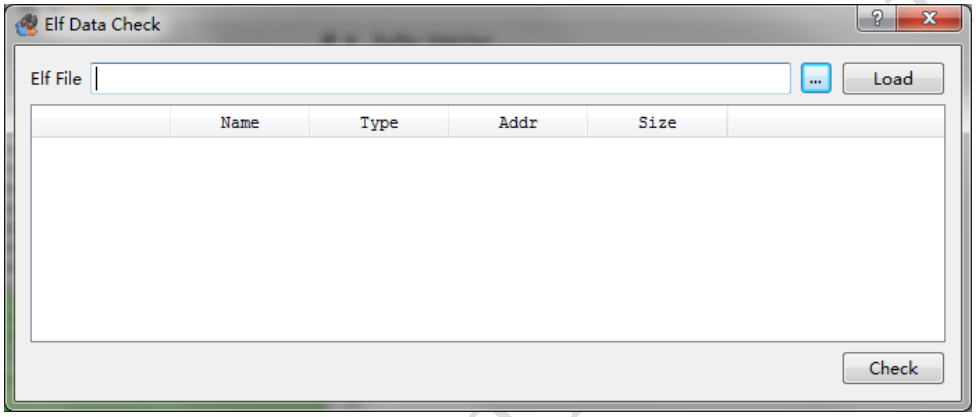


Figure 4-7-11 Elf Data Check interface

Operation Steps

1. Click <load> button, choose the *.elf file, click <Check> button.

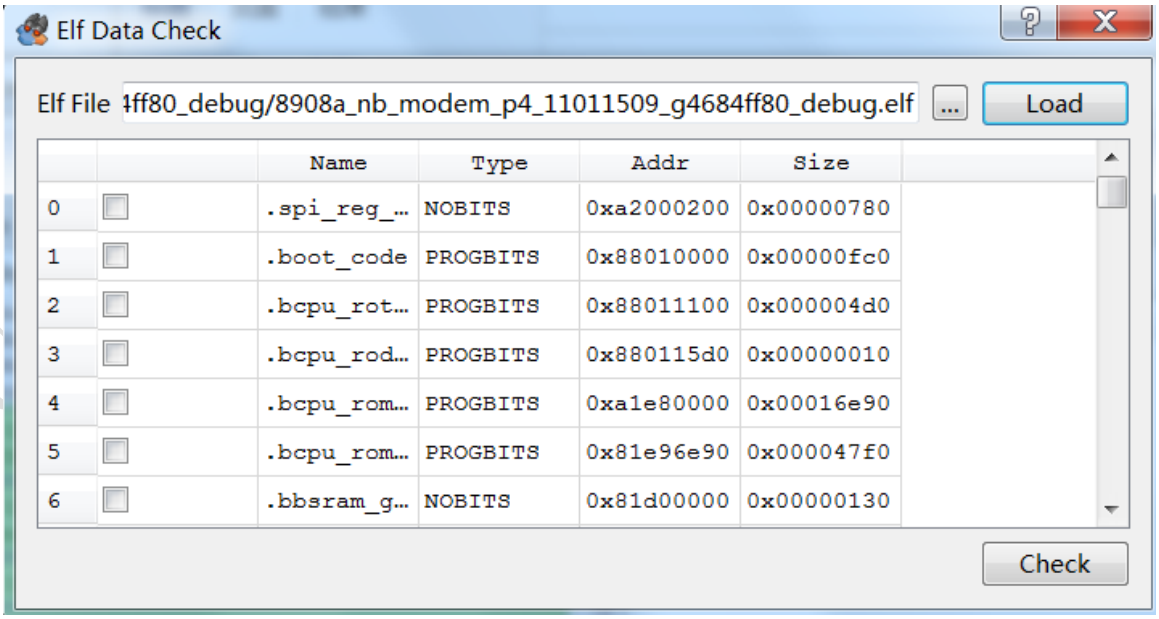


Figure 4-7-12 load elf

2. Choose the items to compare

Choose the items to compare by the check box.

5	✖	.bcpu_rom_ro…	PROGBITS	0x81e8e960	0x00002700
6		.bbsram	PROGBITS	0x81980000	0x00001400
7		.bbsramu	PROGBITS	0xa1981400	0x00000d40
8	✖	.bbsram_glob…	PROGBITS	0x81983260	0x00000190

Figure 4-7-13 select compared items

3．Compare and save result

Click the <Check> button in Figure 4-7-12. If with difference, will pop up a dialog.

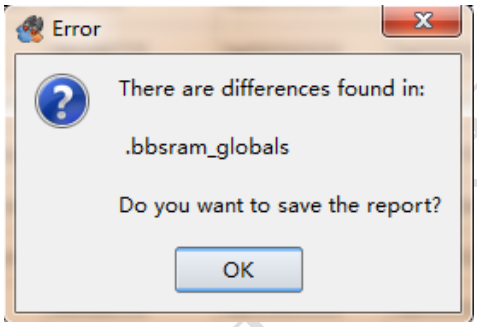


Figure 4-7-14 error information

Click <OK>, enter the file name to save.

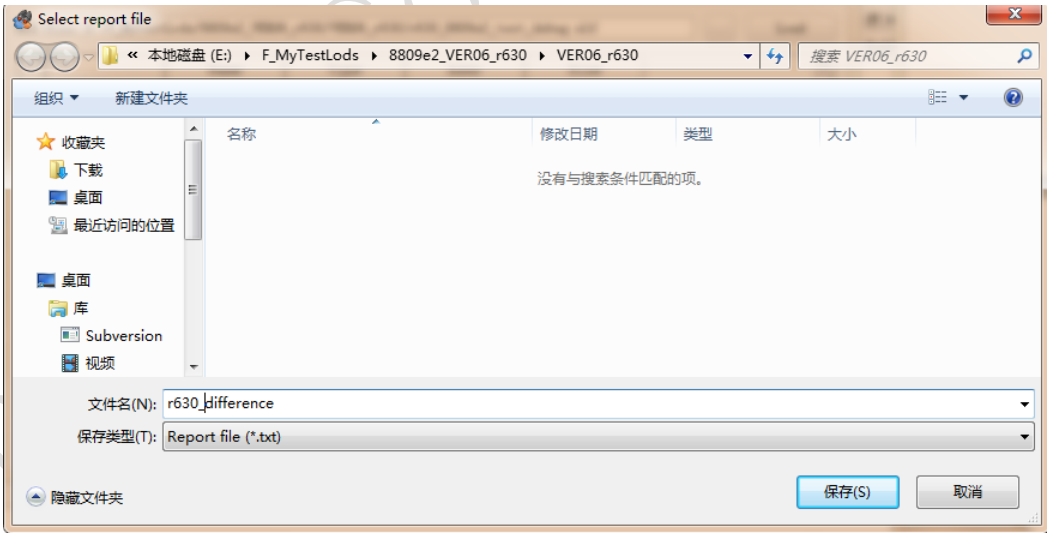


Figure 4-7-15 save report file

The difference information will be saved to the file.

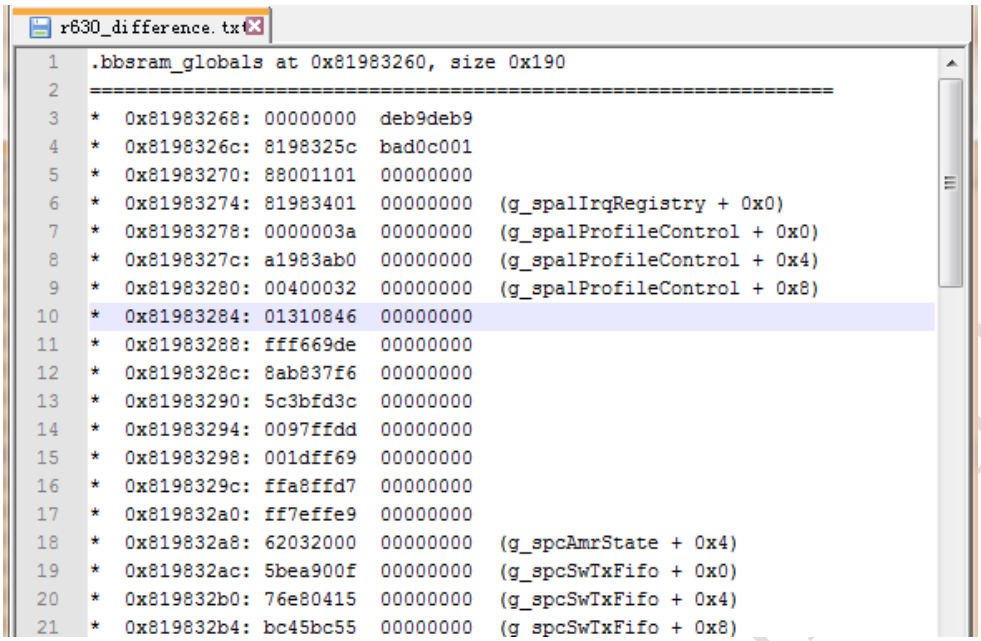


Figure 4-7-16 Comparison of different parts in the report

4.8 Heap Report

HeapReport function has two plugins, SX Heap Report and Heap Report (New).

Different module use different plugin, 8910 should use Heap Report (New).

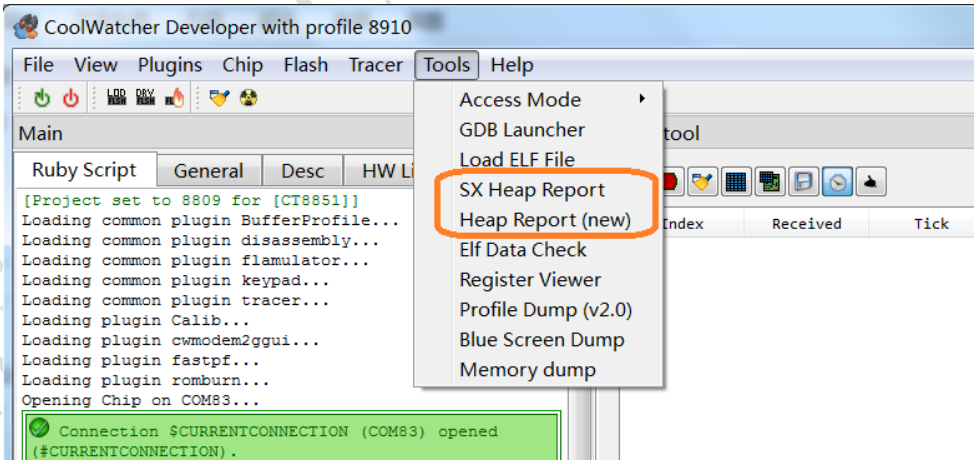


Figure 4-8-1 Heap Report menu

4.8.1 Save Heap Report File

4.8.1.1 8910 and Following Modules

8910 and following modules should use Heap Report (New).

Shown in Figure 4-8-1, click main menu <Tools> —> < Heap Report (New)>, activate Heap Report (New) dialog, shown as Figure 4-8-2.

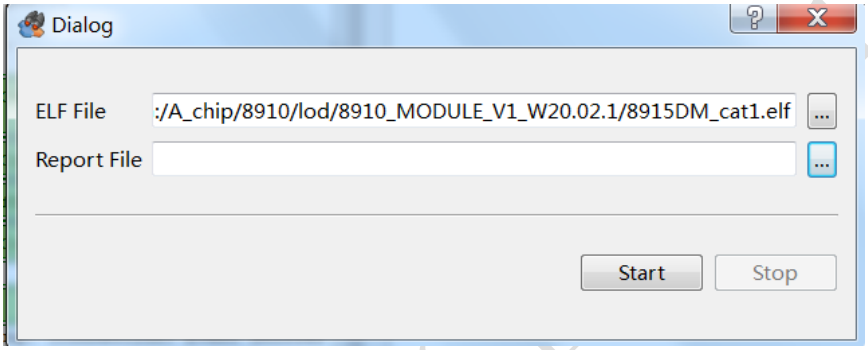


Figure 4-8-2 activate Heap Report (new)

Choose correct ELF file and file name to be saved. The ELF file is the save version with published lod. Click start button, when Figure 4-8-3 is displayed, the dump is succeed.

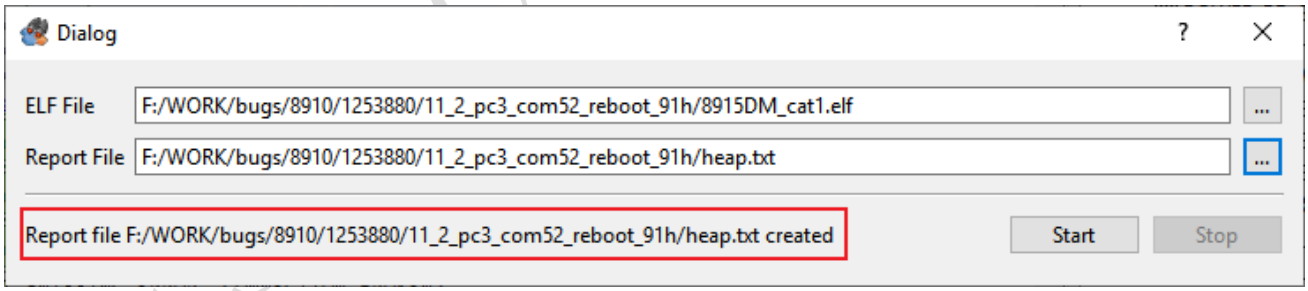


Figure 4-8-3 Heap Report (New) dump succeed

4.8.1.2 8955/8908/8909 .etc

8955/8908/8909 .etc. modules should use SX Heap Report.

Shown in Figure 4-8-1, click main menu <Tools> —> <SX Head Report>, activate Heap Report dialog, shown as Figure 4-8-4.

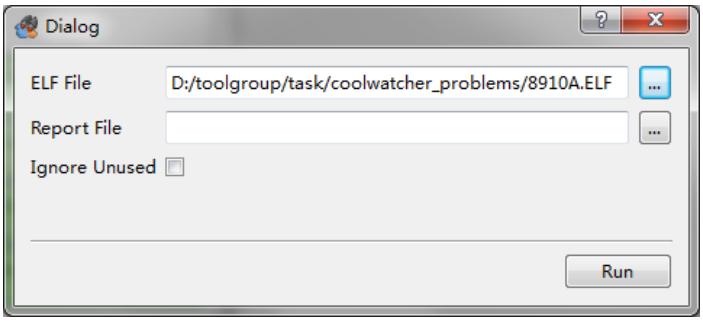


Figure 4-8-4 Heap Report dialog

Elf file: published with lod of the save version.

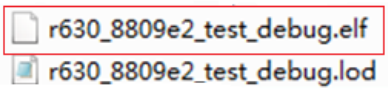


Figure 4-8-5 select elf file

Report File: the result file.

Enter or select file name, click <Run> to get Report File.

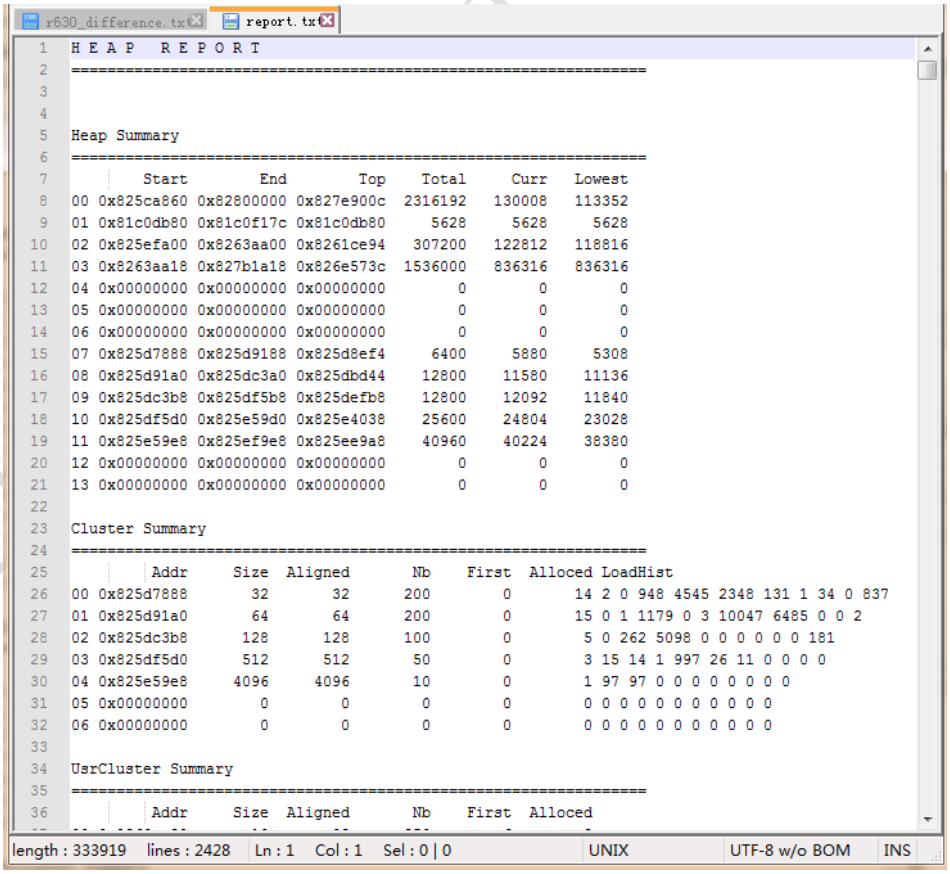


Figure 4-8-6 Report File

4.8.2 Heap Report Analyze

Heap report analysis do not has some special skills, experience is very important. If a function malloc a large memory and do not release, then memory leak may exist. Saved report file is different between 8955/8908/8909 and 8910, but recording information are most the same.

4.9 Register Viewer

RegisterViewer is a tool that can read by bit, write register, and search functions.

Click main menu <Tools> —> <Register Viewer>.

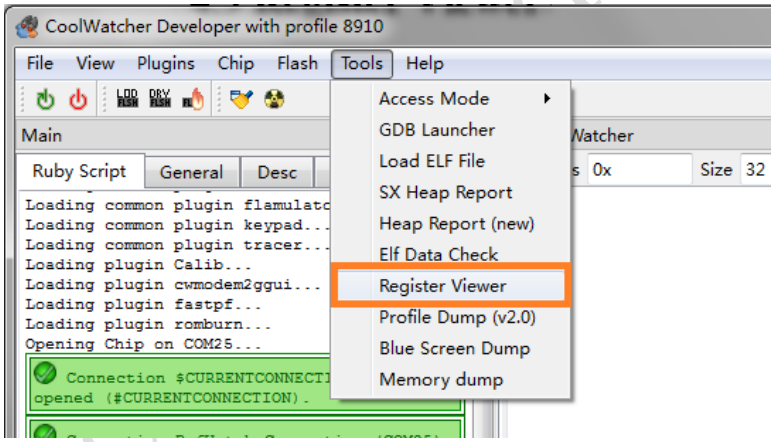


Figure 4-9-1 start Register Viewer

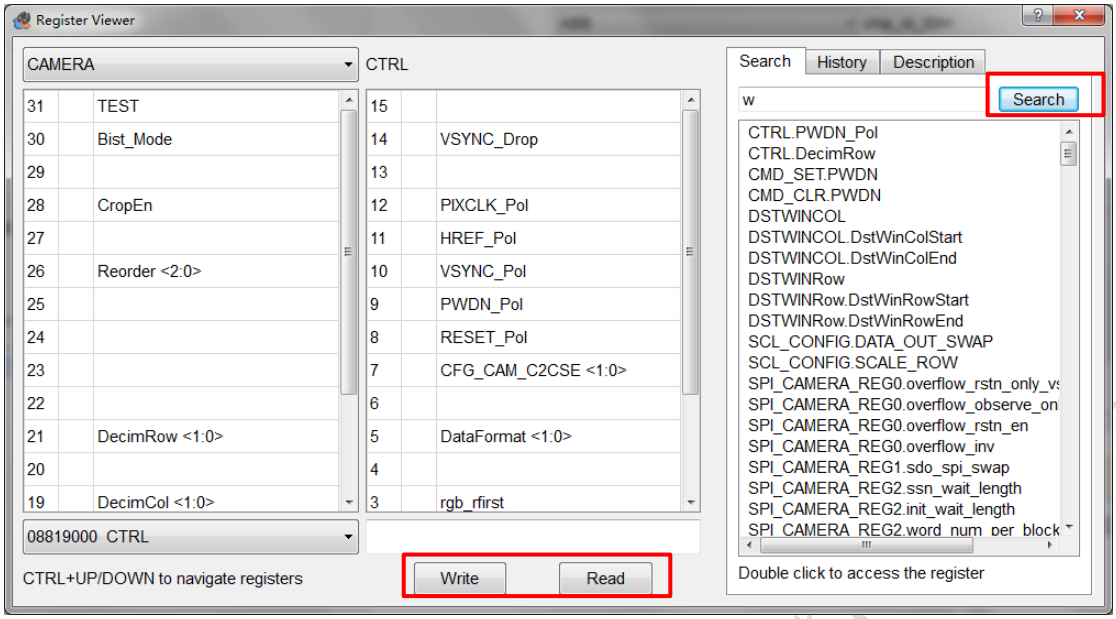


Figure 4-9-2 Register Viewer interface

4.10 Blue Screen Dump

4.10.1 Start Blue Screen Dump

4.10.1.1 8910 and Following Modules

8910 and following modules use Blue Screen Dump tool.

Click main menu <Tools> —> <Blue Screen Dump>, the main interface shown as Figure 4-10-2.

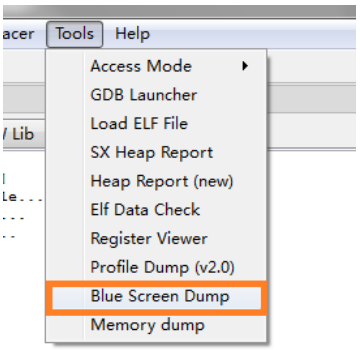


Figure 4-10-1 start Blue Screen Dump

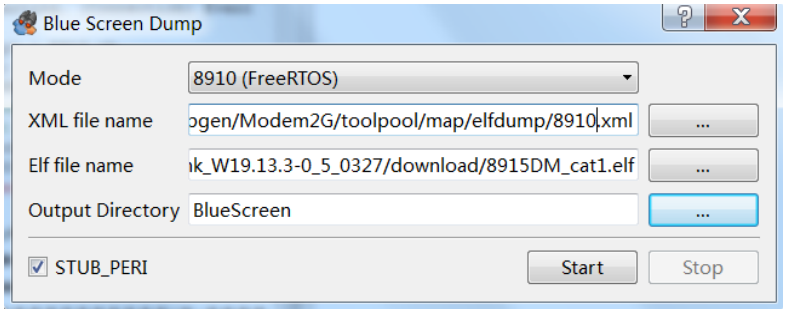


Figure 4-10-2 Blue Screen Dump interface

Settings:

Five Mode, ARM dump bin only、MIPS dump bin only、8910(FreeRTOS)、8909(FreeRTOS)、8955(FreeRTOS), can be selected for different chip. 8910->8910(FreeRTOS).

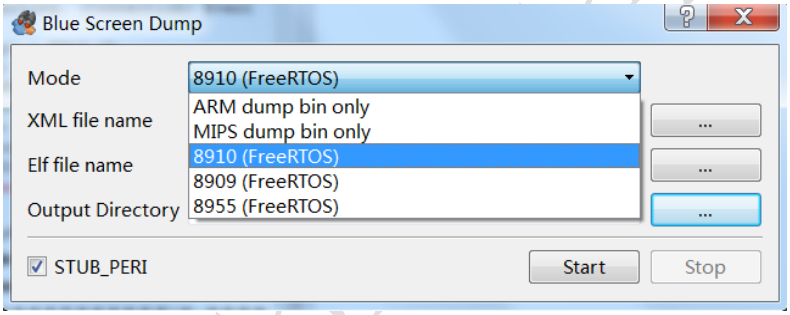


Figure 4-10-3 five Mode of Blue Screen Dump

XML: select default elfdump xml file.

Output Directory: path to save dump file, should select new path every time.

STUB_PERI: this is fast dump mode (software stub,with ADI/ISPI/RFSPI), default checked ☑ STUB_PERI. If this is checked, fast dump mode will be try first, when fast dump fail, the original dump mode would be try again.

Note:

If dump fail, the commad “r 0” should be run first to ensure the read function is fine. if 如 read fail, run “creconnet” command first , then “r 0”. If read function is fine but dump fail, “STUB_PERI” should be set as unchecked status ☐ STUB_PERI.

Blue screen dump for 8910 does not to parse elf file any more.

Log information

Dumped log will be saved in the same folder of dump *.bin file, named as dump_*.log.

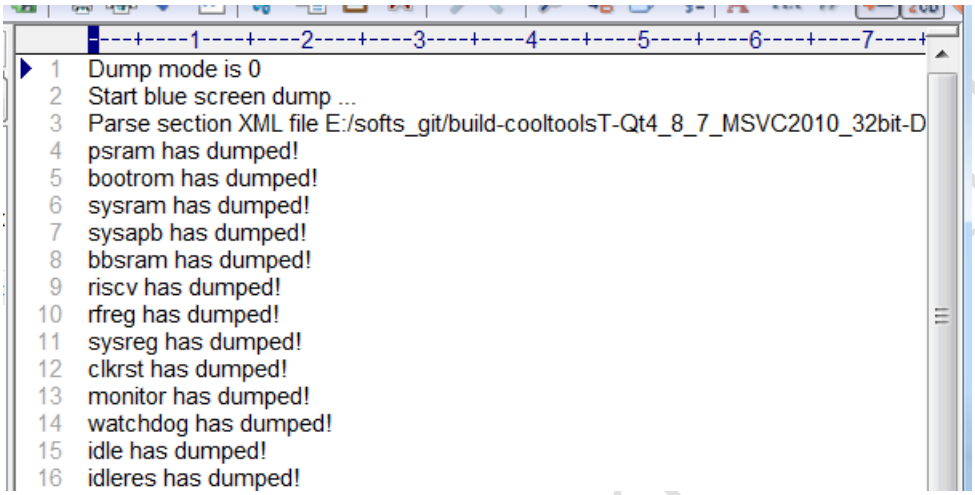


Figure 4-10-4 Blue Screen Dump log information

4.10.1.2 8955/8908/8909 .etc

8955/8908/8909 .etc. modules should use comand "elfdump" after chip is hang, such as Figure 4-10-5.

```
> elfdump "r630.elf"
building elf for 8809e2...
Reading page reg @0x81a0c000...
done
Reading xcpu reg @0x81a2b000...
done
Reading bcpu reg @0x8190a000...
done
```

Figure 4-10-5 use elfdump command

4.10.2 Blue Screen Analyze

1. 8910 blue screen analyze

   - Analyze AP

   Copy the AP elf file to dump path, name as ap.elf.

Drag the ap.cmm into TRACE32, then start to analyze.

- Analyze CP

Copy the CP axf file to dump path, name as cp.axf.

Drag the cp.cmm into TRACE32, then start to analyze.

2. Dump register contains PMIC register, and others can be added later.
3. AP FreeRTOS change task cannot work now.
4. Cmm can be loaded to coolhost offline.

Coolhost displays the same when load ap.cmm and cp.cmm.

Coolhost load *.cmm file offline, then activate GDB can also analyze blue screen. But TRACE32 is more recommend.

4.11 Access Mode

Note：Use this function carefully。

Click Tools -> Access Mode to activate this function.

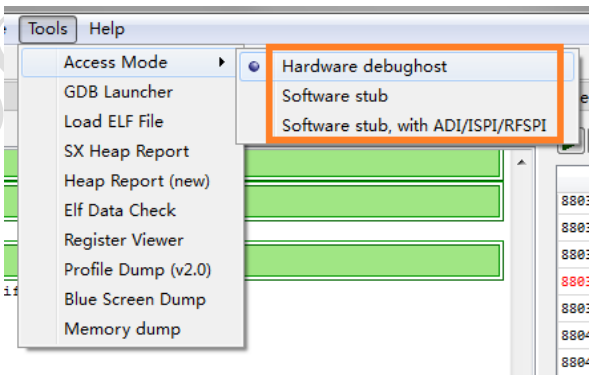


Figure 4-11-1 three mode of Access Mode

Now, coolwatcher has three mode to read and write.

1. DebugHost hardware.
2. Send command to CPU and wait CPU to return back data.

3. Send command to CPU and wait CPU to return back data, at the same time CPU will handle the specific address.

   8910: in the way of handle ADI BUS to read and write PMIC register.

It is suggest that change to PMIC mode only in necessary, change to Hardware DebugHost mode when do not need to read and write PMIC register.

When read and write PMIC register through software, spinlock will be used, so conflicts can be avoided.

4.12 Cmm Generate

If Blue screen dump failed， the *.cmm file does not genereate, but some of *.bin files have been saved in PC, such as psram.bin, then “Cmm Generate” function can be used to generate *.cmm file.

Click Tools -> Cmm Generate to activate this function.

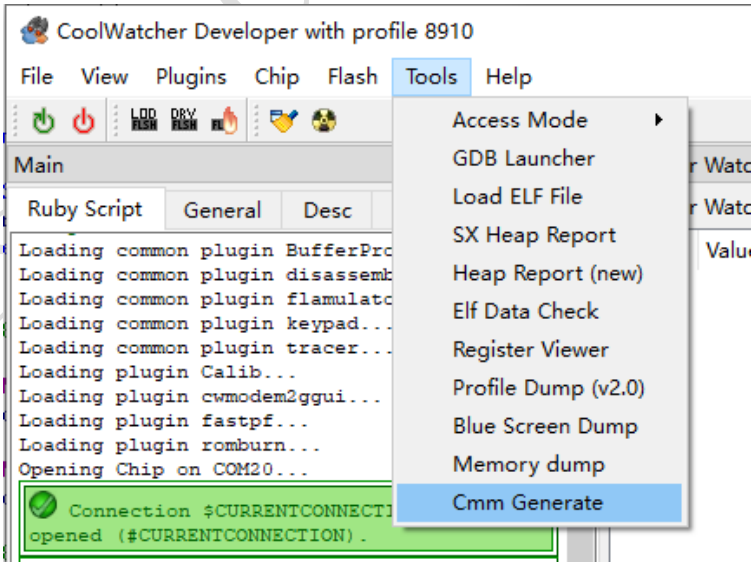


Figure 4-12-1 start Cmm Generate

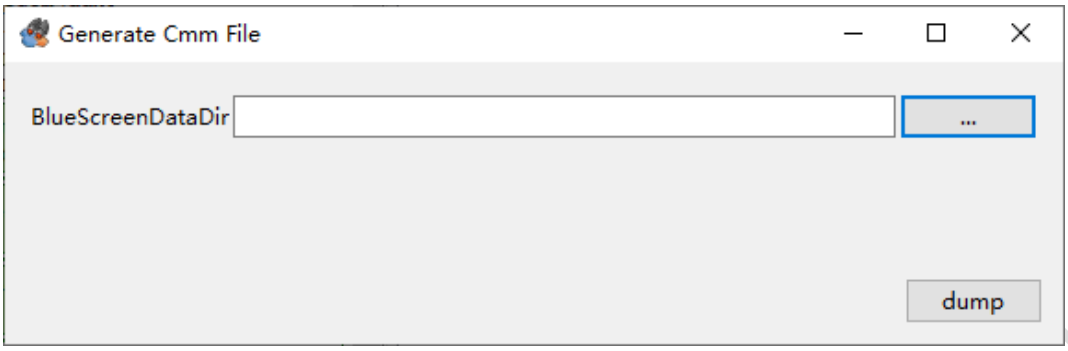


Figure 4-12-2 Generate Cmm File Dialog

Choose the blue screen data director, which saved *.bin files, then click dump button, the*.cmm file will be generate and saved in the chosen director.

4.13 Chip Control

This tools provide the common functions to control the chip that greatly improves the ability to control the phone hardware during the debug process.

4.13.1 Shutdown Ship

Click menu <Chip> —> <Turn off the chip> or button  on toolbar, can shut down UIS8910DM.

4.13.2 Restart Chip

Click button  on toolbar to restart chip.

4.13.3 Force Panic

Click menu <Chip> -> <Force panic> can force chip panic, shown as Figure 4-13-1.

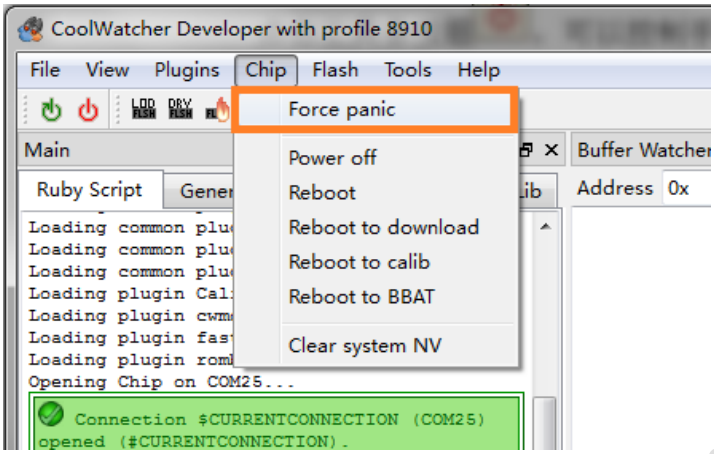


Figure 4-13-1 Force panic

Force panic send command to CPU through DebugUart, if CPU is occupied by other high priority task, the panic will not occur.

4.13.4 Other Chip Option

Shown in Figure 4-13-1, except "Force panic" option, following chip option can also be supported.

- Chip -> Power off,
- Chip -> Reboot,
- Chip -> Reboot to download: reboot and enter ROM download mode.
  - This option do not replace the "FORCE DOWNLOAD PIN", only provide a quicker accessible method. It is recommend to use "FORCE DOWNLOAD PIN".
- Chip -> Reboot to calib: reboot and enter calibration mode.
  - It is recommend to use "SIMBA ENTER MODE".
- Chip -> Reboot to BBAT
  - It is recommend to use "BBAT ENTER MODE".

4.14 Binary Trace Replay

Applicable scene: cannot save trace through coolwatcher, save binary file by other tools, but need to analyze trace through coolwatcher.

Saved trace do not have complete time information, if the interval between two logs is needed, only consulting binary trace is not enough.

Solution as following:

1. Tools to catch package (need to be modified).

Timestamp should be add to trace information, rules as following:

- Auto add a host timestamp package when construct a new log file.
- Auto add a host timestamp package, if the UART do not catch data during a certain interval, e.g. 1 second.
- If UART cannot catch data, add host timestamp package per second.
- Support automatic segmentation function to avoid too large binary file, every file is not large than 10M.

2. Coolwatcher

Timestamp package will be parsed, when bin file is loaded in coolwatcher. Before getting another timestamp, tick is cumulative to record time.

3. Format of host timestamp package

```
struct CH_TIMESTAMP {
        quint8 sync;
        quint8 lenM;
        quint8 lenL;
        quint8 flowid;
        quint32 date;
        quint32 ms;
```

```
        TIMESTAMP (){
            sync= 0xAD;
            lenM = 0;
            lenL = 0x08;
            flowid = 0xa2;
            date = 0;
            ms = 0;
        }
    };
```

- sync、lenM、lenL、flowid: default value, cannot to be modified.
- Date: calculate by “year << 16 + month << 8 + day”.
- ms: millisecond.
- endian: little endian.

4. Coolwatcher replay UART data

Click menu Tracer->Load Trace (bin), then select *.bin file to load.

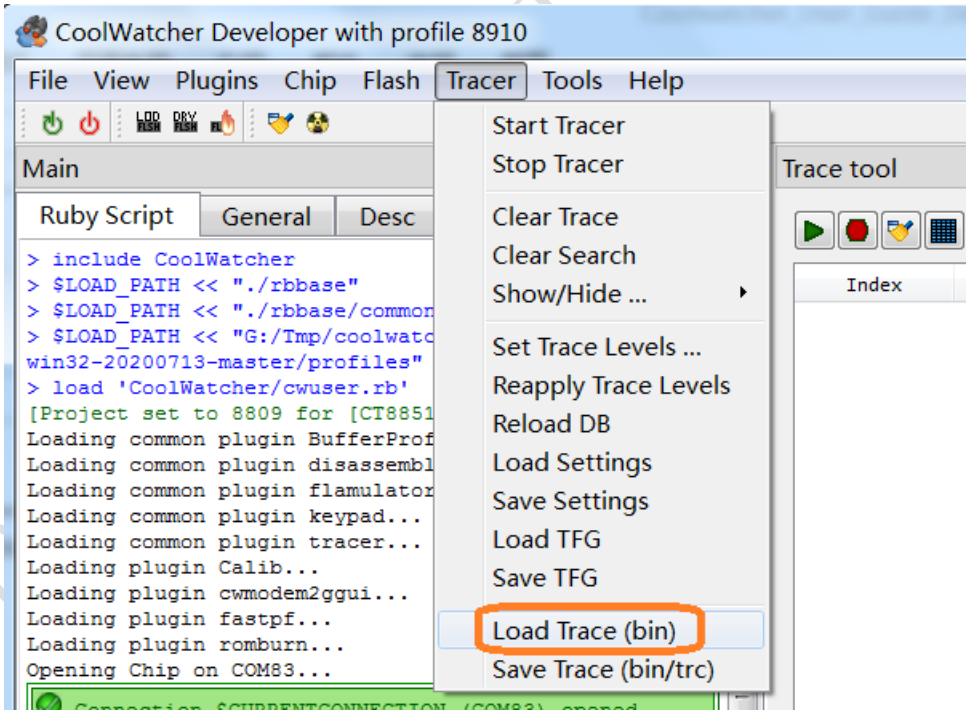
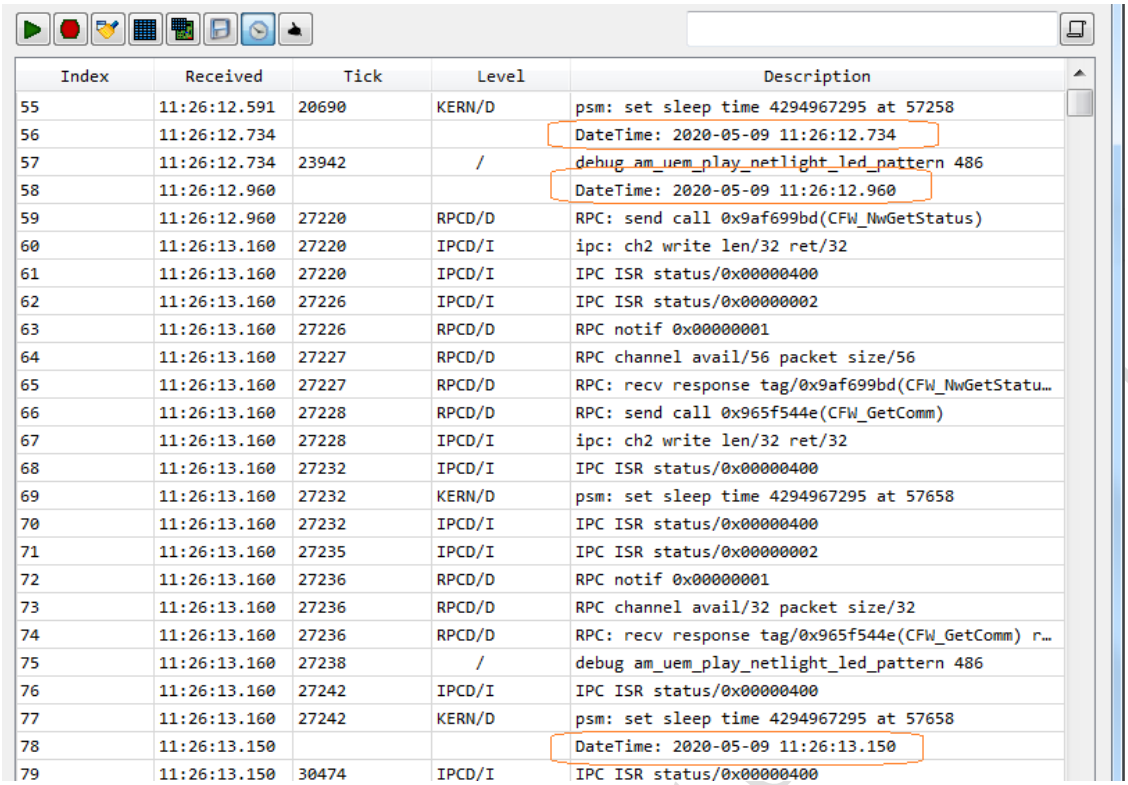


Figure 4-14-1 select UART replay item

When Uart data with timestamp is replayed, result will be displayed as Figure 4-14-2.



Index	Received	Tick	Level	Description
55	11:26:12.591	20690	KERN/D	psm: set sleep time 4294967295 at 57258
56	11:26:12.734			DateTime: 2020-05-09 11:26:12.734
57	11:26:12.734	23942	/	debug am_uem_play_netlight_led_pattern 486
58	11:26:12.960			DateTime: 2020-05-09 11:26:12.960
59	11:26:12.960	27220	RPCD/D	RPC: send call 0x9af699bd(CFW_NwGetStatus)
60	11:26:13.160	27220	IPCD/I	ipc: ch2 write len/32 ret/32
61	11:26:13.160	27220	IPCD/I	IPC ISR status/0x00000400
62	11:26:13.160	27226	IPCD/I	IPC ISR status/0x00000002
63	11:26:13.160	27226	RPCD/D	RPC notif 0x00000001
64	11:26:13.160	27227	RPCD/D	RPC channel avail/56 packet size/56
65	11:26:13.160	27227	RPCD/D	RPC: recv response tag/0x9af699bd(CFW_NwGetStatu…
66	11:26:13.160	27228	RPCD/D	RPC: send call 0x965f544e(CFW_GetComm)
67	11:26:13.160	27228	IPCD/I	ipc: ch2 write len/32 ret/32
68	11:26:13.160	27232	IPCD/I	IPC ISR status/0x00000400
69	11:26:13.160	27232	KERN/D	psm: set sleep time 4294967295 at 57658
70	11:26:13.160	27232	IPCD/I	IPC ISR status/0x00000400
71	11:26:13.160	27235	IPCD/I	IPC ISR status/0x00000002
72	11:26:13.160	27236	RPCD/D	RPC notif 0x00000001
73	11:26:13.160	27236	RPCD/D	RPC channel avail/32 packet size/32
74	11:26:13.160	27236	RPCD/D	RPC: recv response tag/0x965f544e(CFW_GetComm) r…
75	11:26:13.160	27238	/	debug am_uem_play_netlight_led_pattern 486
76	11:26:13.160	27242	IPCD/I	IPC ISR status/0x00000400
77	11:26:13.160	27242	KERN/D	psm: set sleep time 4294967295 at 57658
78	11:26:13.150			DateTime: 2020-05-09 11:26:13.150
79	11:26:13.150	30474	IPCD/I	IPC ISR status/0x00000400

Figure 4-14-2 replay Uart data with timestamp

4.15 Command Line Operation

This tools provide some common commands. User can input these commands in the command line to run the operation. And the result will show in the “Rubby script” line.

4.15.1 COM Port Operation

The common port commands list in the table.

5.

command	parameter	e.g.	note
copen	(NUM,BR=BR_AUTOMATIC)	copen(2,115200)	Open COM2
reop		reop	Reopen port

4.15.2 Flash Programming

The common flash programming commands list in the table.

command	parameter	e.g.	note
fastpf	(flash_programmer_filename, lod_filename, disable_event_sniffer = true)	fastpf("c:\xxxx_ramrun.lod" ," " c:\\lod")	Burn Flash
fastSectorEraser	(flash_programmer_filename, sector_list, disable_event_sniffer = true)	fastSectorEraser("c:\xxxx_ramrun.lod" ," [0x01000000,0x01200000] ")	Erase sector

4.15.3 Read Flash

The common read flash commands list in the table.

command	parameter	e.g.	note
r	(addr)	r x01004000	32 bit write
r32	(addr)	r32 0x01004000	32 bit write
r16	(addr)	r16 0x01004000	16 bit write
r8	(addr)	r8 0x01004000	8 bit write

4.15.4 Write Flash

The common write flash commands list in the table.

command	parameter	e.g.	note
w	(addr,val)	w(0x01004000,0xffffffff)	32 bit write
w32	(addr,val)	W32(0x01004000,0xffffffff)	32 bit write
w16	(addr,val)	W16(0x01004000,0xffff)	16 bit write
w8	(addr,val)	W8(0x01004000,0xff)	8 bit write

4.15.5 Other Commands

command	parameter	e.g.
elfdump	elf file [,xml file] xml file is not necessary. If add xml, give the absolute path	elfdump “test.elf” or elfdump “test.elf” , “D:/8809.xml”
dump	(str_filename,address,nbwords), read a range of continuous flash content to the specified file	Dump(“c:\\xxxx.lod” ,0x01004,0x100)
chipID	none, return chip id directly	
flashReadStatus	noe	
flashSectorErase	flashSectorErase(addr)	
flashBlockErase	flashBlockErase(addr)	
flashBlock32kErase	flashBlock32kErase(addr)	
flashChipErase	flashChipErase()	
xcvRead	xcvRead(addr)	

xcvWrite	xcvWrite(addr,data)	

4.16 Other Functions

4.16.1 Kill Command Thread

Click <File> —> <Kill command thread> or button  on toolbar, can terminate the running task (the thread was started by the command line).

4.16.2 Kill Running Thread

Click <File> —> <Kill all thread>, can stop all the running tasks.

4.16.3 Clear Screen

Click main menu <View> —> <Clear Screen> or button  on toolbar, can clear the output message at the bottom-left area of Ruby script.

4.17 Notice

4.17.1 Read/Write Register

INT_REG_DBG_HOST register cannot be read in 8910 module.

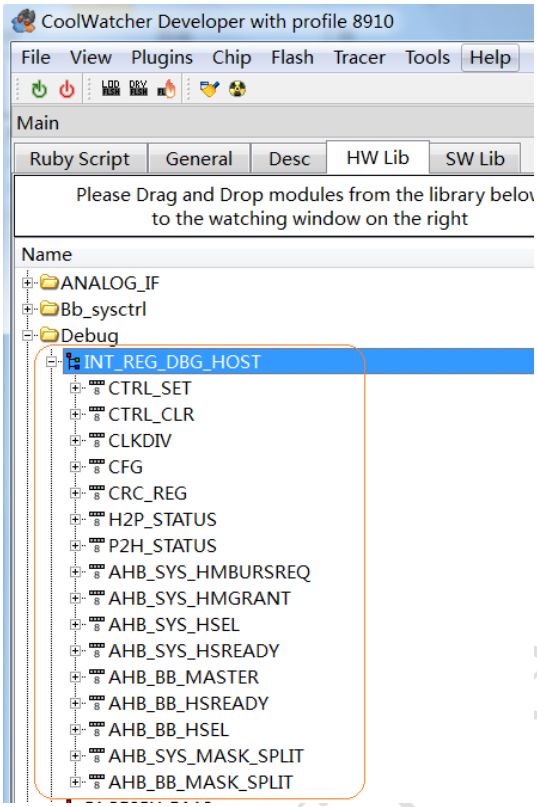


Figure 4-17-1 8910 INT_REG_DBG_HOST register

4.17.2Change Connection Type for 8910DM

USB and debughost two connection types are provided to be selected when 8910DM connect with coolwatcher. And this two connections type can be switched.

Notice for connection switch:

- Burn flash trough coolwatcher is not support, please use reseachdownload to burn flash. And only debughost connection supports burn operation.
- USB switch to debughost: When 8910DM module connects through coolwatcher_usb.exe, and chip under panic status, connection type can be changed to debughost. Blue screen dump and GDB server can worked very well through debughost connection.

- Debughost switch to USB: When 8910DM module connects through coolwatcher_debughost.exe, and chip under panic status, connection type can be changed to USB. But "Unisoc usb serial port" must be enumerated before chip panic, or usb connection cannot established. After change to usb connection, blue screen type can be used directly in coolwatcher, and GDB server can be used through "DTools.exe" with "UART" connection.