

Unisoc Coolwatcher

User Guide

Release Date	2020/03/03
Document No.	
Version	V0.5
Document Type	Word
Platform	8910/8909L 等
OS Version	Windows XP 及 以 上,ubuntu16.04

声明 Statement

本文件所含数据和信息都属于紫光展锐所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负任何与本文件相关的直接或间接的、任何伤害或损失。

All data and information contained in or disclosed by this document is confidential and proprietary information of UNISOC and all rights therein are expressly reserved. This document is provided for reference purpose, no license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document, and no express and implied warranties, including but without limitation, the implied warranties of fitness for any particular purpose, and non-infringement, as well as any performance. By accepting this material, the recipient agrees that the material and the information contained therein is to be held in confidence and in trust and will not be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without the express written permission of UNISOC. UNISOC may make any changes at any time without prior notice. Although every reasonable effort is made to present current and accurate information, UNISOC makes no

guarantees of any kind with respect to the matters addressed in this document. In no event shall UNISOC be responsible or liable, directly or indirectly, for any damage or loss caused or alleged to be caused by or in connection with the use of or reliance on any such content.

Unisoc Confidential

关键字 Keywords

环境配置、烧写、buffer、寄存器、GDB、离线分析、Heap Report、芯片控制、命令行操作、

蓝屏 dump

Unisoc Confidential

版本历史 Revision history

版本 Version	日期 Date	作者 Author	描述 Description
V0.2	2016-03-05	Tool-Group	初始版本
V0.2	2017-09-18	Tool-Group	完善文档
V0.3	2019-07-25	Tool-Group	修改文档格式
V0.4	2019-10-15	Tool-Group	8910 USB 用法及注意事项
V0.5	2020-03-03	Tool-Group	修改查看 T race 章节

前 言

一 范围 Scope

本文详细介绍了 Coolwatcher 的基本功能与使用说明，适用于 Coolwatcher 使用人员。

二 内容定义 Details Definitions

1. 定义 Definitions
2. 符号定义 Symbols
3. 缩略语 Abbreviations

无

三 参考文献 References

无

目 录 Contents

声明 Statement.....	2
关键字 Keywords.....	4
版本历史 Revision history	5
前 言	6
1. 概览 Overview	10
1.1 文档概要	10
1.2 目标设定	10
2. 环境配置介绍	10
2.1 软件安装.....	10
2.2 运行环境.....	10
2.3 License 问题.....	10
2.4 启动软件.....	11
3. 安装与设置	11
3.1 启动 Coolwatcher	11
3.2 Coolwatcher 在 linux 下的使用 (Ubuntu16.04)	14
4 功能说明	15
4.1 烧写 Flash.....	15
4.2 查看 buffer 信息	17
4.3 查看寄存器.....	17
4.4 查看 Trace.....	18
4.4.1 启动 tracer	18
4.4.2 Tracer 菜单项.....	19

4.4.3 Tracer 应用步骤.....	21
4.5 GDB Launcher	23
4.5.1 启动 GDB.....	24
4.5.2 8910 分析调试 ARM.....	26
4.5.3 GDB 常用命令.....	27
4.6 Buffer Profile.....	28
4.6.1 抓起 prf 文件.....	28
4.6.2 分析 prf 文件.....	30
4.7 离线分析.....	31
4.7.1 生成*.core(*.elf).....	31
4.7.2 建立离线分析环境	32
4.7.3 离线分析 GDB.....	34
4.7.4 Elf Data Check.....	35
4.8 Heap Report	38
4.8.1 SX Heap Report.....	38
4.8.2 Heap Report (new)	39
4.9 Register Viewer.....	40
4.10 Blue Screen Dump.....	41
4.10.1 启动 Blue Screen Dump.....	41
4.10.2 蓝屏分析	43
4.11 Profile Dump	44
4.12 Access Mode	44

4.13	芯片控制	45
4.13.1	关闭芯片	45
4.13.2	重新启动芯片	46
4.13.3	芯片强制死机	46
4.13.4	其他 chip 操作	46
4.14	8910 USB 用法及注意事项	47
4.15	命令行操作	50
4.15.1	端口操作	50
4.15.2	Flash 编程	51
4.15.3	读 Flash	51
4.15.4	写 Flash	52
4.15.5	其他命令	52
4.16	其他功能	53
4.16.1	Kill 当前运行的程序	53
4.16.2	Kill 所有运行的程序	53
4.16.3	清除脚本输出信息	53
4.17	Linux 串口配置	53
5	Q & A	54
6	附录 Appendix	54

1. 概览 Overview

1.1 文档概要

本文档详细介绍了 Coolwatcher 的基本功能与使用说明，包括硬件配置、软件配置、使用步骤、故障处理等。

1.2 目标设定

Coolwatcher 工具使用人员，可以熟练、正确使用 Coolwatcher 的各项功能，以便于为后续分析、解决问题提供帮助。

2. 环境配置介绍

2.1 软件安装

Coolwatcher 是绿色软件，无需安装，将压缩包解压到某个子目录下即可使用。注意解压路径中，不要含有中文字符。

2.2 运行环境

建议计算机配置内存 4G 以上、处理器双核以上，需确保安装相应的串口驱动，以保证模块/手机可与 PC 成功连接并通讯。

可支持的 PC 系统版本包括 Windows XP/7/10 等，linux 系统仅支持 ubuntu16.04。

2.3 License 问题

Coolwatcher 无需授权可直接使用。

2.4 启动软件

双击软件目录中的 Coolwatcher.exe 文件，即可启动软件。

3. 安装与设置

双击打开 coolwatcher.exe 的配置界面，如图 3-1 所示。左侧为模块/手机型号选择区，右侧为配置项。

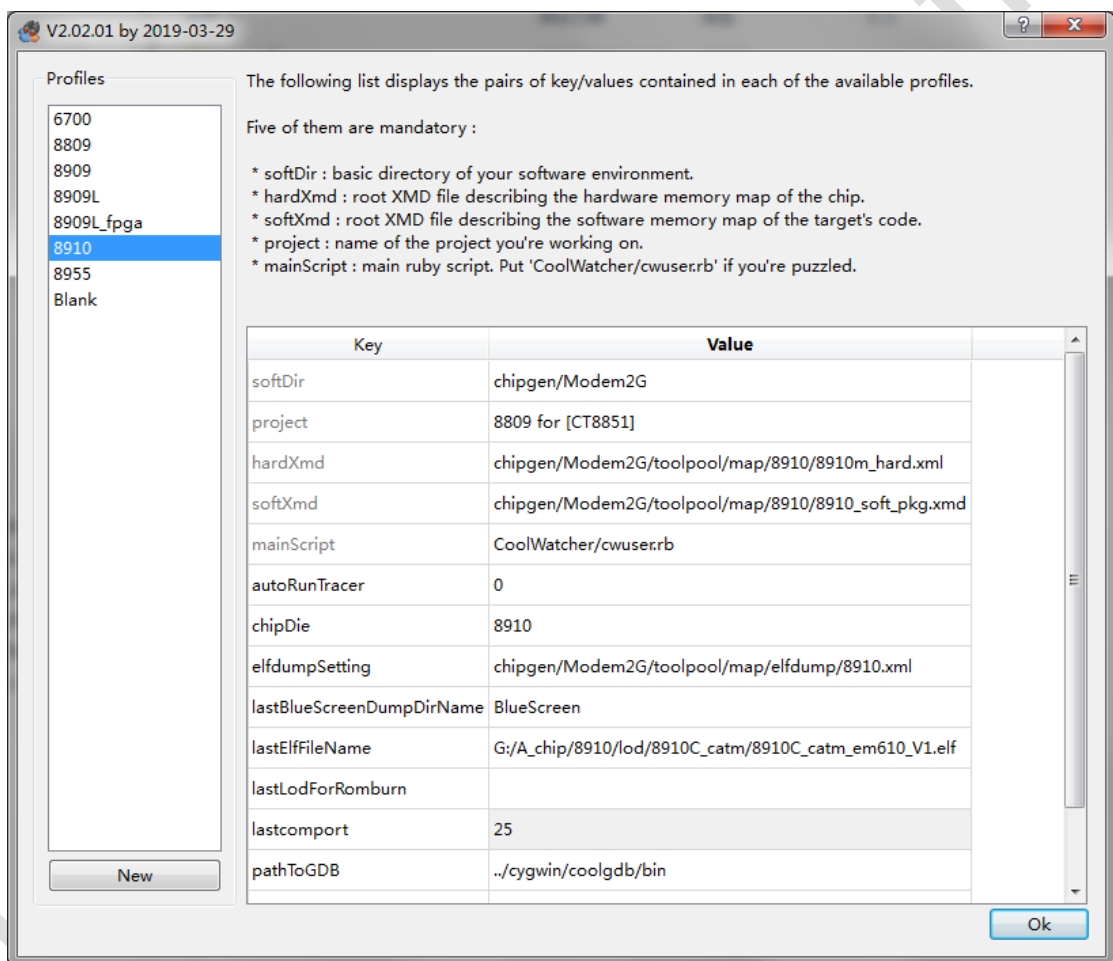


图 3-1 Coolwatcher 配置界面

3.1 启动 Coolwatcher

1. 根据模块/手机类型选择 Profiles 类型，如 UIS8910DM 选择 8910；
2. 修改必要的配置项：如端口号；

单机“OK”键，“coolhost”配置界面将会被打开 界面如图 3-2 所示。选择相应的 COM 口，之后单击“connect”。

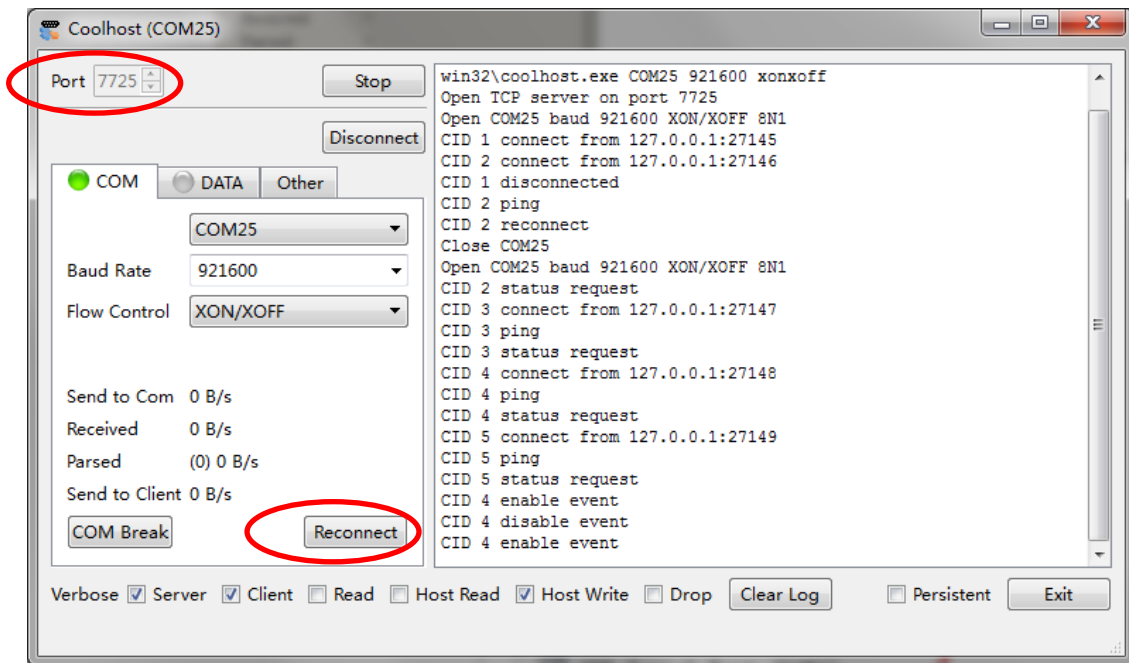


图 3-2 coolhost 界面

端口号可通过“计算机->管理->设备管理器->端口”进行查看，以 UIS8910DM 为例，如图 3-3 所示。UIS8910DM 连接 Coolwatcher 选择设备识别出的第二个 COM 口（此处为 COM25）

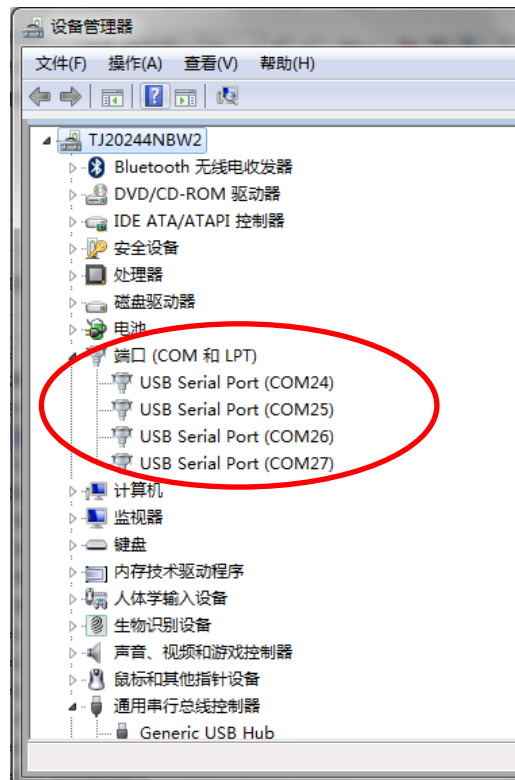


图 3-3 查看端口

Windows 下，端口范围建议用 0~255，如果端口号过大，可通过端口属性修改端口号；
linux 中，软链接建立的端口号需为 1~255 之间，即 COM1、COM2 ... COM255。

3. Coolwatcher 主界面

Coolwatcher 主界面如图 3-4 所示。

主界面包括：菜单栏、工具栏、log 打印区、HW Library 显示树、SW Library 显示树，寄存器查看区，Buffer 读取区等功能区域。

Coolwatcher 工具可实现烧录 flash、检查 buffer 信息、读取寄存器、查看 trace 以及 dump 数据等功能。

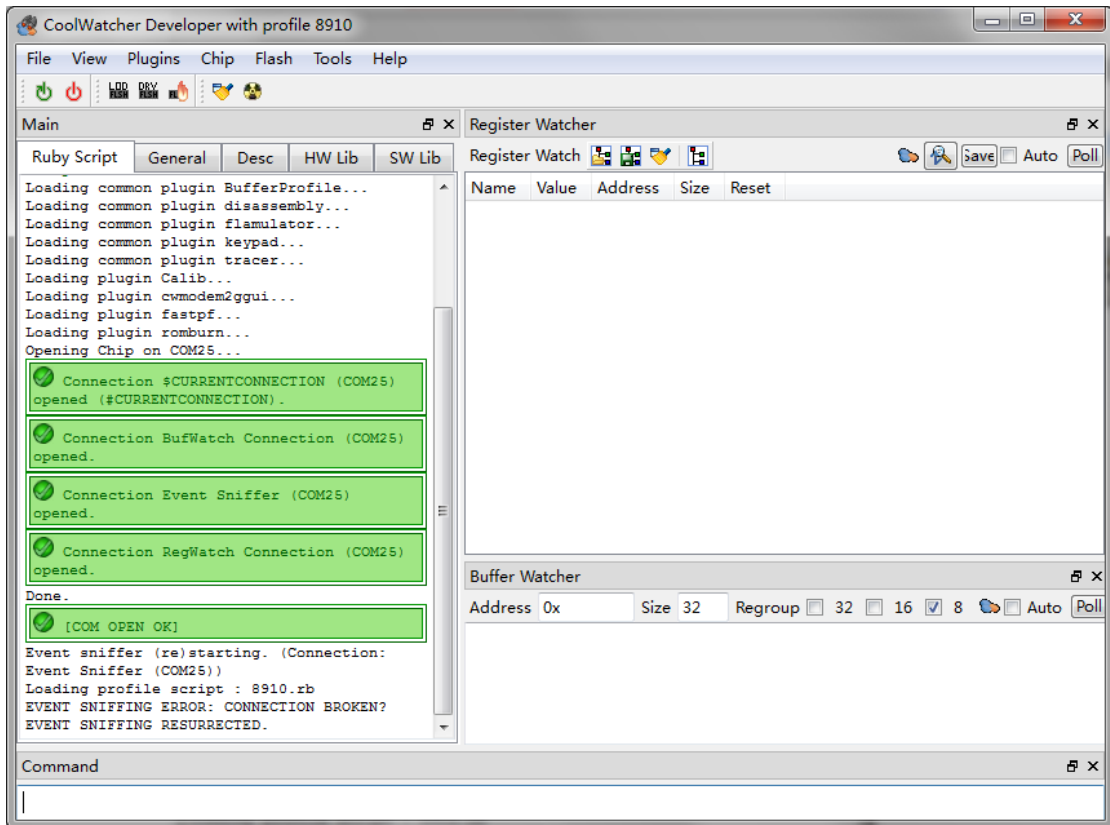


图 3-4 coolwatcher 主界面

3.2 Coolwatcher 在 linux 下的使用 (Ubuntu16.04)

1. 依赖包安装

用 `sudo apt install` 命令逐个安装 build-essential、libqt4-qt3support、itcl3、itk3、iwidgets4

2. 99-coolsand-dongle.rules

把 99-coolsand-dongle.rules 文件放到 /etc/udev/rules.d/路径中

3. 端口配置

在应用程序 coolwatcher.exe 所在文件夹中建立子文件夹 comport。

为/dev/ttyUSB0、/dev/ttyUSB1 创建符号连接文件

如 `ln -s /dev/ttyUSB0 comport/COM1`

`ln -s /dev/ttyUSB1 comport/COM2`

访问端口。

注意 COM1、COM2 字母需大写。


4 功能说明

4.1 烧写 Flash

注意：该功能不支持 8910，8910 烧写使用 ResearchDownload 工具。

烧写 Flash 需要以下几个步骤：

1. 将手机或模块通过 dongle 正确连接到 PC 的 USB 端口上。
2. 选择 lod 文件。(注：lod 文件不要放在有中文字符的路径下面)

点菜单项 “Flash->Choose Lod file” 或工具条按钮 ，选定要下载到 Flash 内的 Lod 文件（如图 4-1、4-2）。

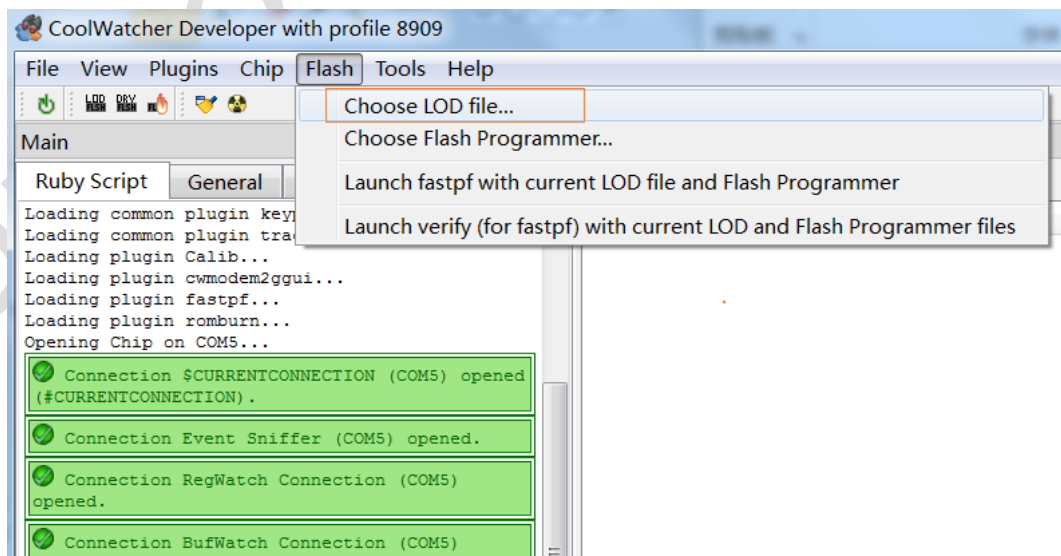


图 4-1 选择 lod 文件

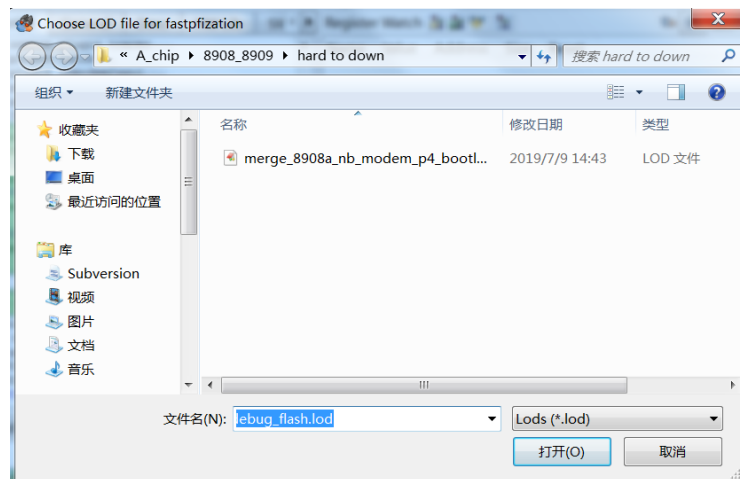



图 4-2 选择 lod 文件

3. 选择 ramrun 文件

点菜单项 “Flash->Choose Flash programmer...” 或工具条按钮 ，为该手机所使用的 Flash 类型选定正确的 Flash programmer (*_ramrun.lod) 文件（如图 4-3）。

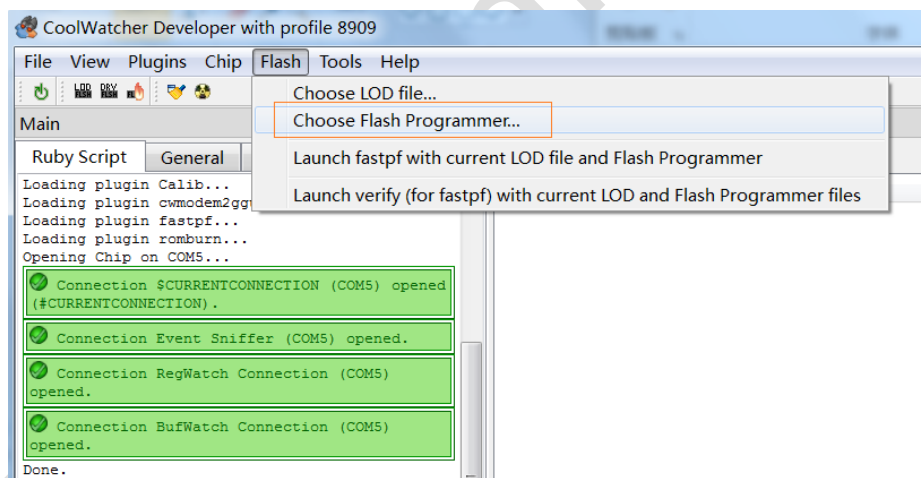



图 4-3 选择 ramrun 文件

4. 开始烧录

单项 “Flash->Launch fastpf with current LOD file and flash programmer” 或工具条按钮 ，开始下载 lod file。此时程序右下角出现一个进度条显示下载进度（如图 4-4），下载完成后进度条显示 100%，并且 “Ruby Script” 区显示完成信息。

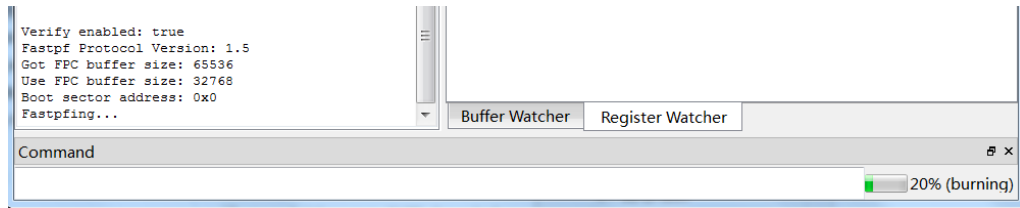


图 4-4 烧录过程中进度条显示

4.2 查看 buffer 信息

在 Buffer Watcher 中的 Address 填写地址，Size 中填写大小，点击 Poll 按钮，可以从模块读取相关数据。

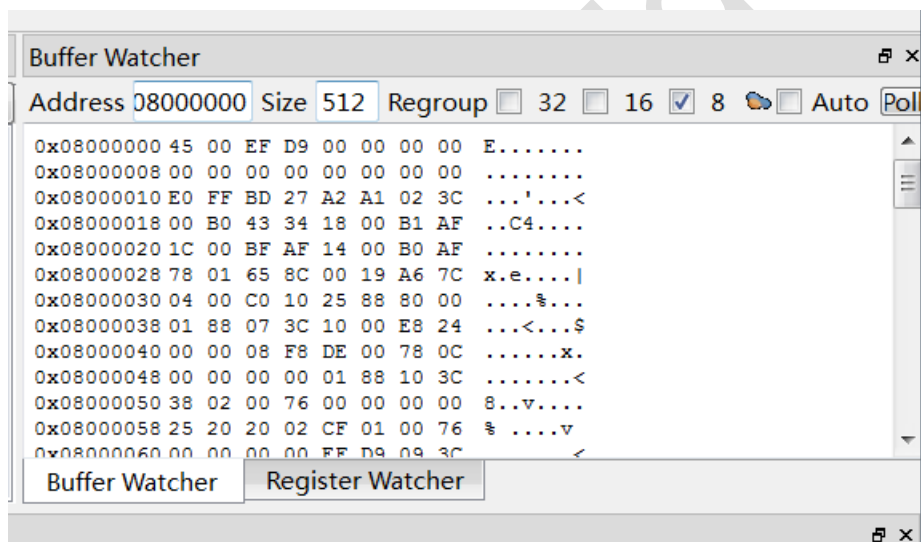


图 4-5 Buffer watcher 窗口

4.3 查看寄存器

把 HW Library 或 SW Library 中的某个节点拖入 Register Watcher 中，点击 pull 按钮，可以读取并显示相关内容。

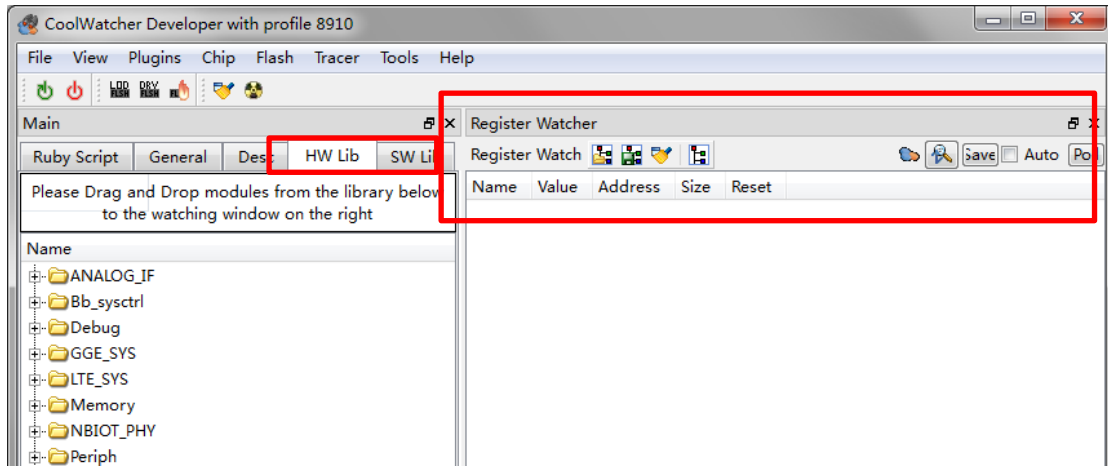


图 4-6 查看寄存器

4.4 查看 Trace

4.4.1 启动 tracer

点击菜单 Plugins->Activate Tracer，如图 4-7 所示，启动 Trace 插件，Trace 主界面如图 4-8 所示。

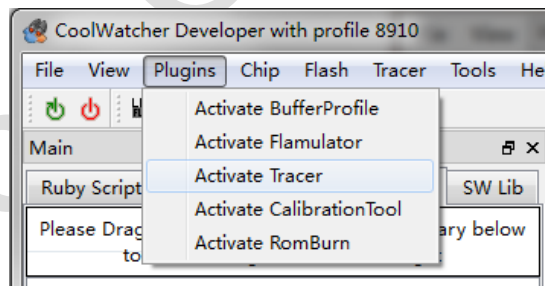


图 4-7 启动 tracer

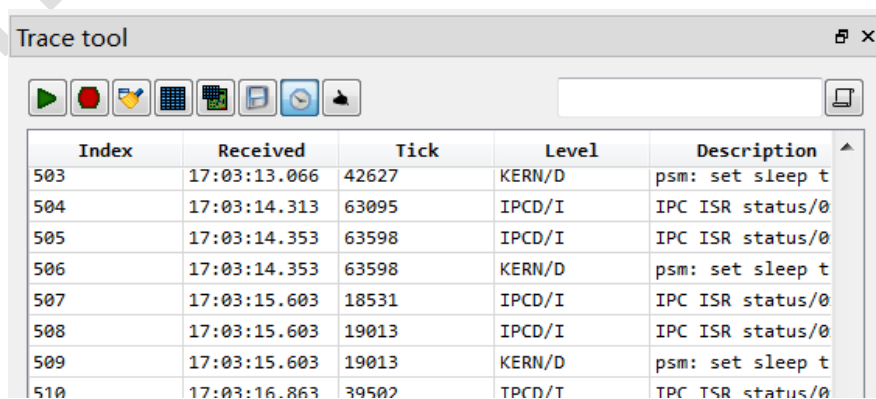


图 4-8 Tracer 主操作界面

工具栏各按钮分别对应 :开始 Trace、停止、清空、设置 TraceLevels、Reapply trace levels、保存、启动/关闭 Received 列、启动/关闭 comment。

各列分别对应 : Trace 序列号、PC 接收 Trace 时间、Tick、Level、描述等。

4.4.2 Tracer 菜单项

插件启动后，主菜单将显示 Tracer 项，如图 4-9 所示:

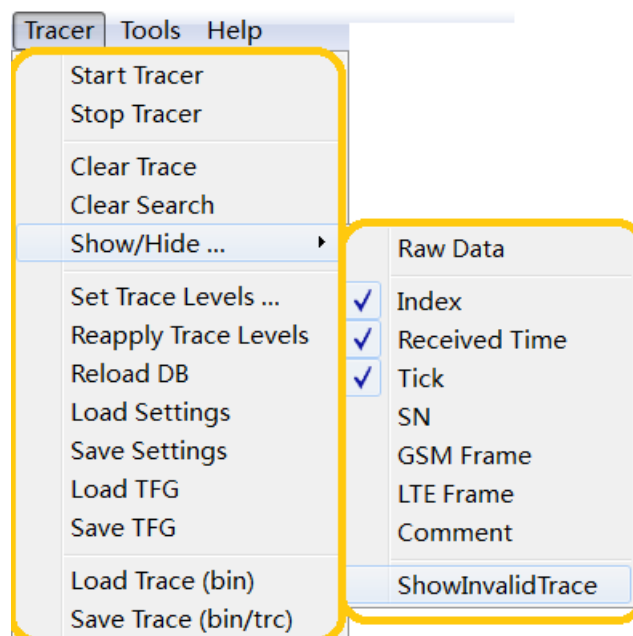


图 4-9 tracer 菜单项

菜单项

Start Tracer : 开始 ;

Stop Tracer : 停止 ;

Clean Trace : 清空 Trace 表格 ;

Clean Search : 清空 Search 记录 ;

Show/Hide ... : 显示或隐藏 tracer 主操作界面相关功能列 ;

Set Trace Levels : 设置 Trace Levels ;

Reapply Trace Levels : 重新应用 Trace Levels ;

Reload DB : 重新加载 DB 文件 ;

LoadSettings : 加载 Levels 配置 ;

SaveSettings : 保存 Levels 配置 ;

Load TFG : 加载 tfg 文件 ;

Save TFG : 选择保存 T 卡 Trace 文件 ;

Load bin: 加载二进制 Trace 文件 , 进行 l o g 回放 ;

Save Trace (bin/trc) : 保存 trace。目前保存的 trc 格式有些变化 , 加了是否采用的 TRACE ID , 可以帮助检查没有使用 TRACE ID 的情况。

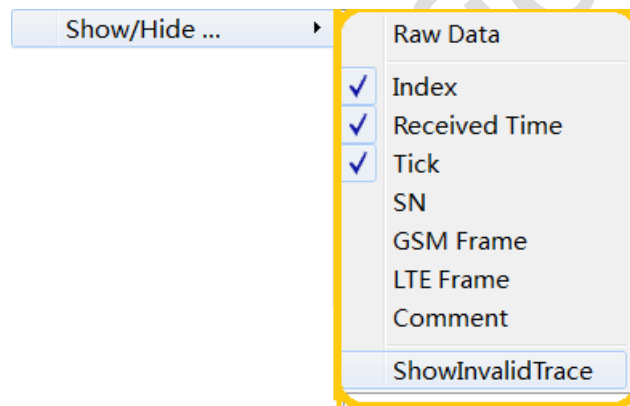


图 4-10 Tracer Show/Hide 菜单

当图 4-10 中选项被勾选时 , 则在图 4-8 的主界面中增加相应的一列 , 来显示相关内容。

菜单选项 :

Toggle Raw Data: 启动/关闭 Raw 表格 ;

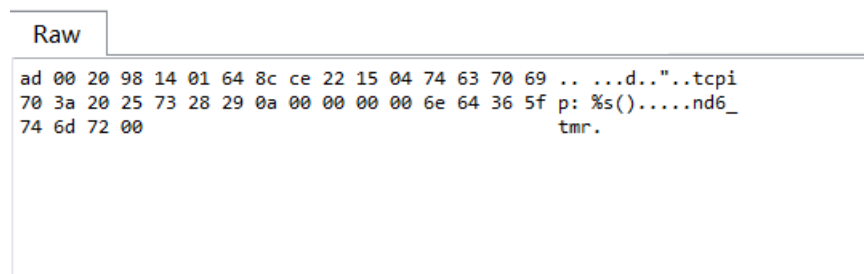


图 4-10-1 R a wData 窗口

Index : Trace 序列号 ;

Received Time : PC 接收到 trace 的时间 ;

Tick : 各 trace 对应的 tick , 显示的 tick 只有 16bits , 16KHz ;

SN : SN 是 16bits 序列号 , 可以用来判断丢 trace 的情况 , 当 trace 没有丢失时 SN 列为空。


ShowInvalidTrace 启动 / 关闭无效 trace 表格. 如下图 , 包括没有使用 TraceID、解析失败、S N 有中断等三个表格。

UnusedTraceID		ParseFailed	SeqJump	
Index	Received	Tick	Level	Description
13	19:30:11.085	11187	KERN/I	ram heap start/80C

图 4-10-2 无效 trace 窗口

4.4.3 Tracer 应用步骤

1. 设置 TraceLevels

点击  按钮 , 设置 tracelevel , 显示如图 4-11。需在左侧表格中选择关注的 levels ;

AutoSave

- check 是否自动保存 trace , 状态为 checked , 则自动保存。
- bin、trc : 为 trace 文件类型 , 前者为二进制文件 , 后者为文本文件;
- Split Size : trace 文件大小。当文件 size 超过该值时 , 自动切分文件 ;

DB file name

DB 文件名。

RowLimit

图 4-8 的 Trace 表格的最大行数；

Auto reapply trace levels on reset

重启前的 TraceLevels 配置信息是否用于重启后。

Tick in flow ID 0x80

该配置项需与 lod 保持一致，如果 lod 中有时间戳，则选中本项，否则不选。

ReceiveEvent

是否接收 Event。

Save Pcap

是否保存 Pcap 信息。

左下角

Save 按钮：保存 Levels 配置；

Load 按钮：加载 Levels 配置；

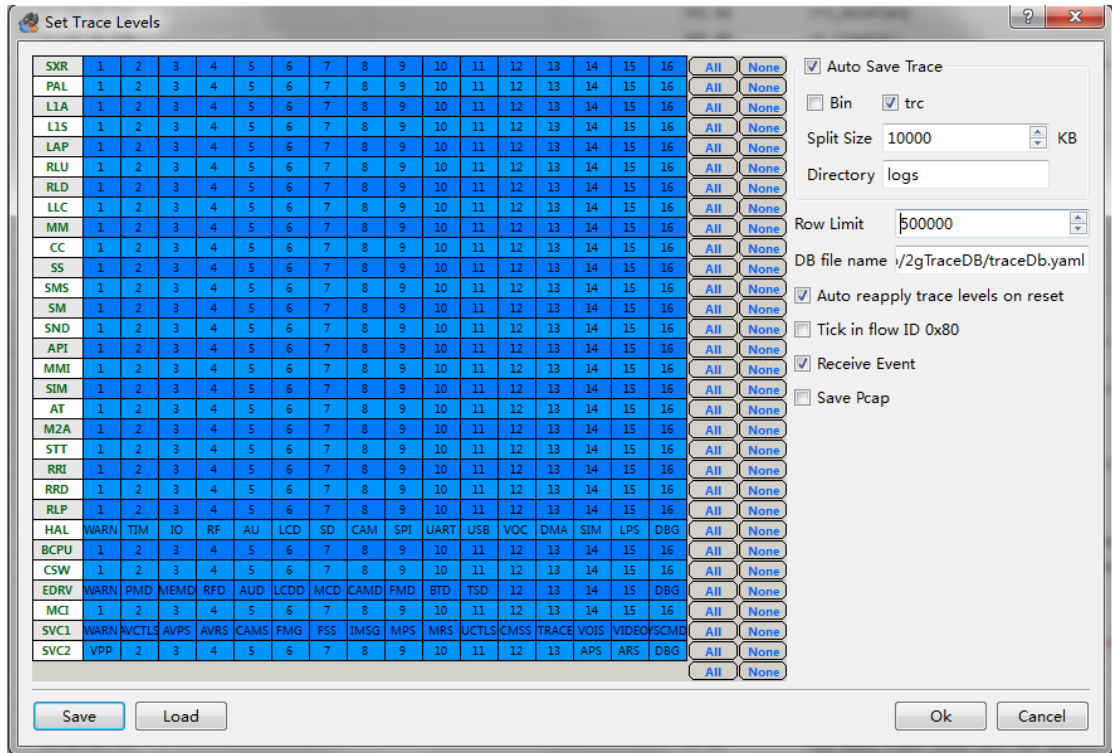


图 4-11 trace level 界面设置

2. 点击图 4-12 中的  按钮，开始 Trace 功能，Trace 信息会显示到表格中。

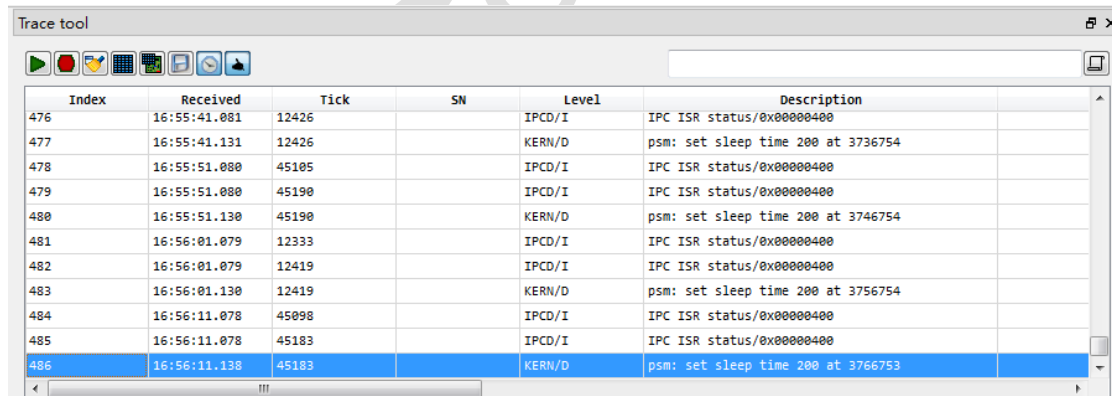


图 4-12 tracer 显示界面

3. 点击图 4-12 中的  按钮，结束 Trace 功能；

注意：Tracer 的配置信息，默认为 rbbase\common\plugins\tracer\ 文件夹中的文件。

4.5 GDB Launcher

GDB 是分析死机、跟踪问题的一种重要方法。

4.5.1 启动 GDB

点击菜单项 Tools->GDB Launcher , 启动如下配置框：

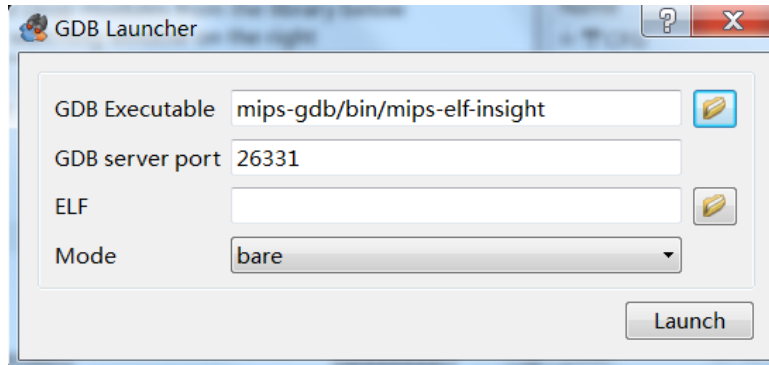


图 4-13 启动 GDB

注意：

- GDB exe 文件：选择 mips-gdb\bin 中的 mips-elf-insight.exe 文件；
- elf 文件：elf 文件对应的 lod 需要与手机/模块中已烧录的 lod 保持一致，一般是与 lod 一起发布的 elf 文件；

- 7 种 Mode：

7 种模式 “bare”、“sx(with REDUCED_REGS)”、“sx(without REDUCED_REGS)”、xcpu_rom、live、8910 AP、8910 CP。

“sx” 相比 “bare” 启动的 gdb 多了两个命令，thread info 和 thread；“xcpu_rom” 一般用于系统启动之前的死机；“live” 模式一般用于严重的死机分析。

8910 模块仅能用 8910 AP 或者 8910 CP 模式，这两种模式不需要选择 GDB Executable 文件。

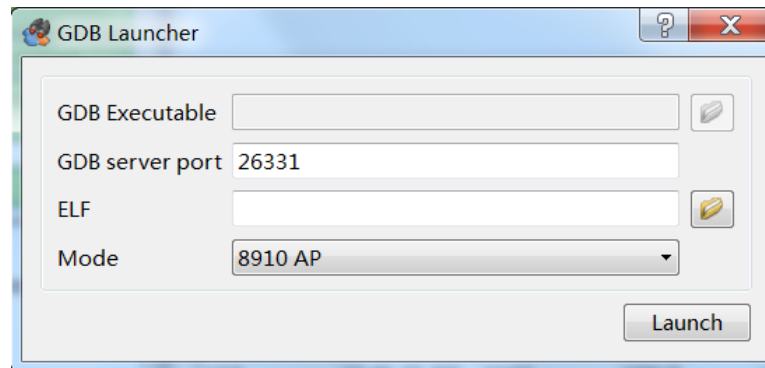


图 4-14 GDB 对话框

点击图 4-13 或 4-14 Launch 按钮后，会弹出如下界面：

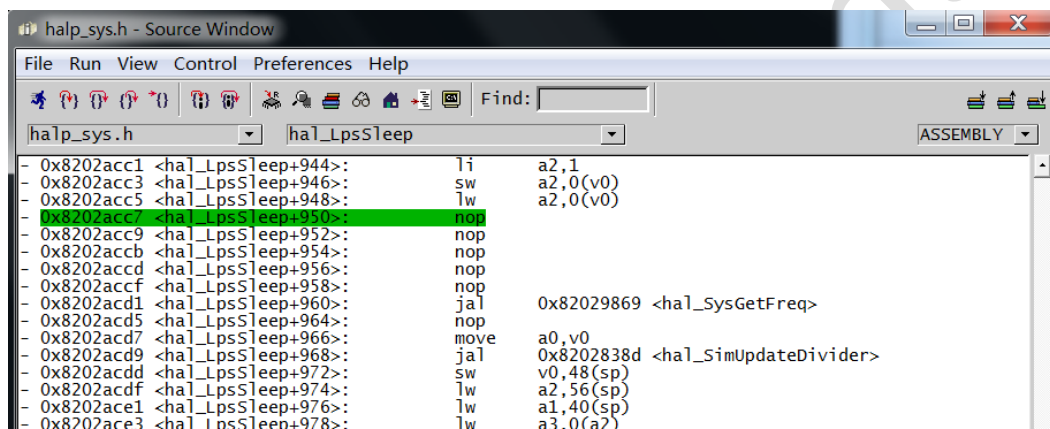


图 4-15 GDB 源码窗口

点击图 4-15 工具栏上的 Console 按钮 ，启动 gdb 命令窗口：

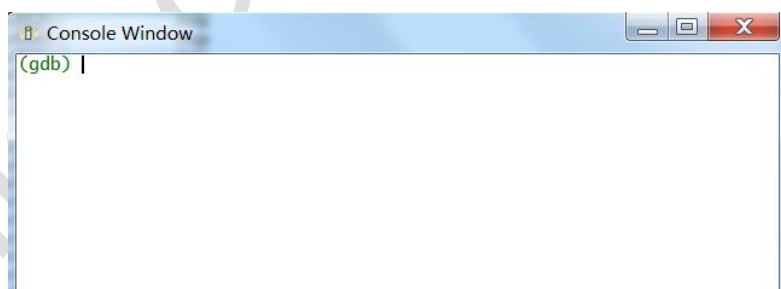


图 4-16 GDB 命令行

输入 gdb 命令，即可分析代码。

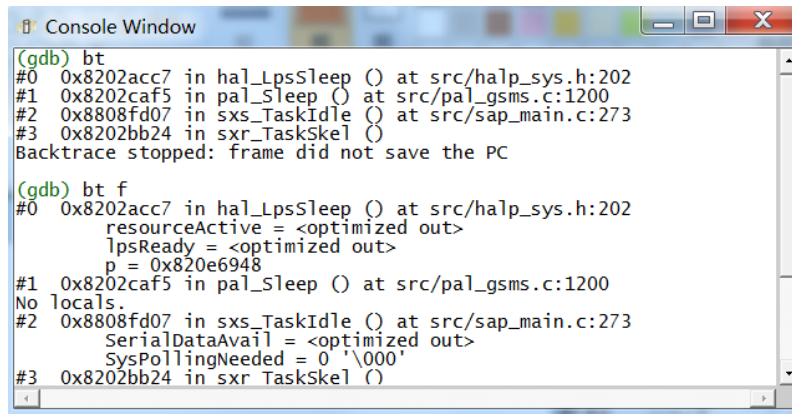


图 4-17 GDB 分析结果显示

4.5.2 8910 分析调试 ARM

GDB Launcher mode 模式选择 8910 AP 或 8910 CP，可以分析调试 8910 ARM。

1. 8910 AP

无需选择 GDB Executable 文件，填写 server port、选择 elf 文件即可；点击 launch 按钮，GDB 会被启动。

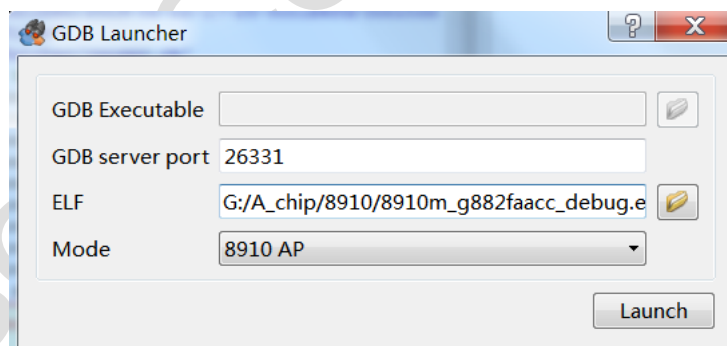


图 4-18 8910 AP GDB 分析

该方式的 mode 为 -cos-abort；

会用到 chipgen/Modem2G/toolpool/gdbfeatures/arm-with-vfpv3.xml 文件。

2. 8910 CP

无需选择 GDB Executable 文件，填写 server port、选择 elf 文件即可；点击 launch 按钮，GDB 会被启动。

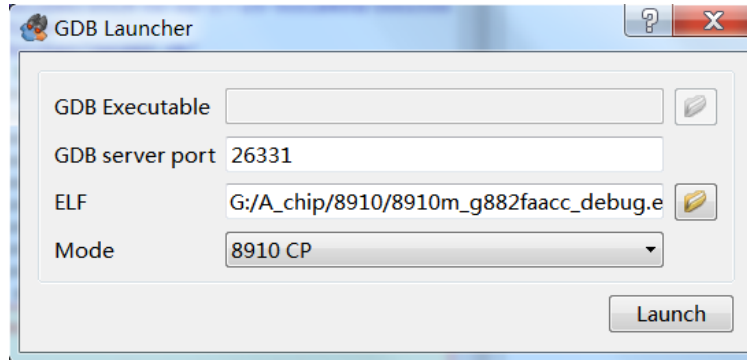


图 4-19 8910 CP GDB 分析

该方式的 mode 为 -8910cp

会用到 chipgen/Modem2G/toolpool/gdbfeatures/arm-core.xml 文件。

4.5.3 GDB 常用命令

ID	GDB 命令	注释
1	p<sth>(print <sth>)	打印 sth 的值, sth 可以是一个表达式、变量、指针等。
2	display <sth>	和 “p sth” 相同,不过 display 会在你每次输入命令后显示 sth 的值。
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 conjunction with bt.

5	down	Goes down into the call stack. To be used in conjunction with bt.

4.6 Buffer Profile

Profile 是分析问题的一种重要方法。

4.6.1 抓起 prf 文件

注意：该方法不能用于 8910 模块，8910 可用 4.10 章节介绍的 Profile Dump 功能。

点击 coolwatcher 主界面中的 Plugins->Activate BufferProfile 菜单项，如图 4-20，启动 Buffer Profile 插件，主菜单上会出现 BufferProfile 菜单，工具栏上会有相关按钮。

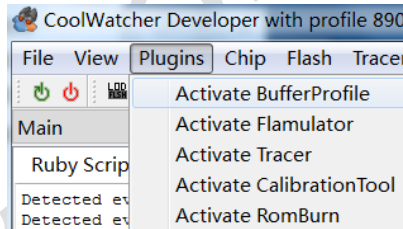


图 4-20 启动 buffer profile

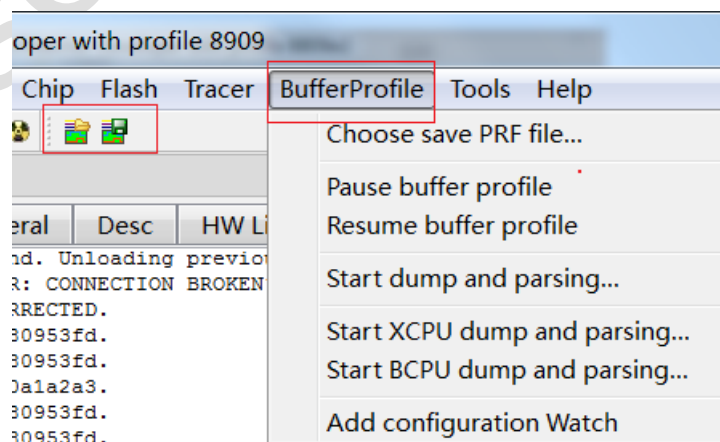



图 4-21 工具栏新增的 bufferProfile 按钮

手机死机后，可以通过该插件抓取有关信息。使用方法如下：

1. 手机通过 COM 口与 PC 机相连，保证可以与 coolwatcher 正常通信。
2. 点击图 4-21 中 Choose save PRF file 菜单项 或工具栏上的  按钮 输入要保存的*.prf 文件，文件的默认路径与 coolwacher.exe 路径一致。
3. 点击图 4-21 中 Start XCPU dump and parsing 菜单，开始保存 XCPU 相关信息。类似信息如下图：

```
> 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
```

图 4-22 保存 XCPU 相关信息

4. 如点击图 4-21 中 Start BCPU dump and parsing 菜单，则保存 BCPU 相关信息。类似信息如下图：

```
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
```

图 4-23 保存 BCPU 相关信息

5. 如点击图 4-21 中 Start dump and parsing 菜单，则保存 BCPU 和 XCPU 两者的信息。

Coolwacher 输出如下图：

```
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
```

图 4-24 保存 XCPU 与 BCPU 信息

注意：

如果手机没有死机，则抓取*.prf 之前，最好点击菜单项中的 Pause buffer profile 项，文件保存完成后，点击 Resume buffer profile 菜单项，恢复状态。

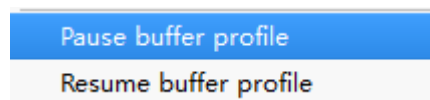


图 4-25 未死机/蓝屏时保存 profile 文件

4.6.2 分析 prf 文件

打开 coolprofile.exe 工具，如图 4-26 所示。

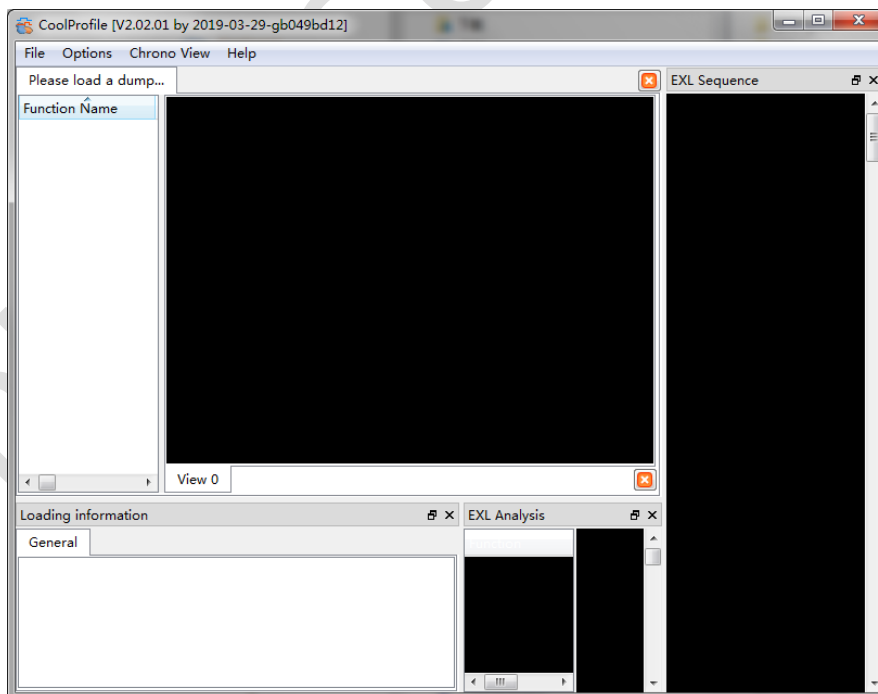


图 4-26 coolprofile 工具主界面

点击 file->open 按钮，通过下对话框，选择 prf 文件。

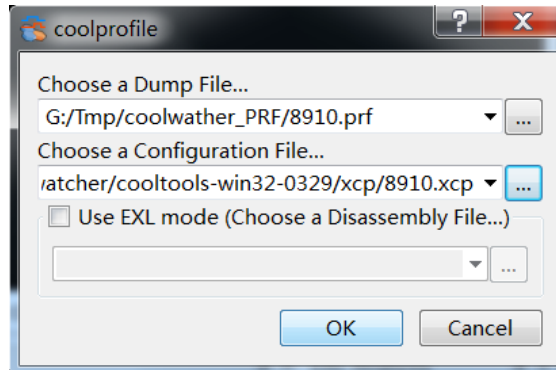


图 4-27 选择 profile 文件

打开效果如下：

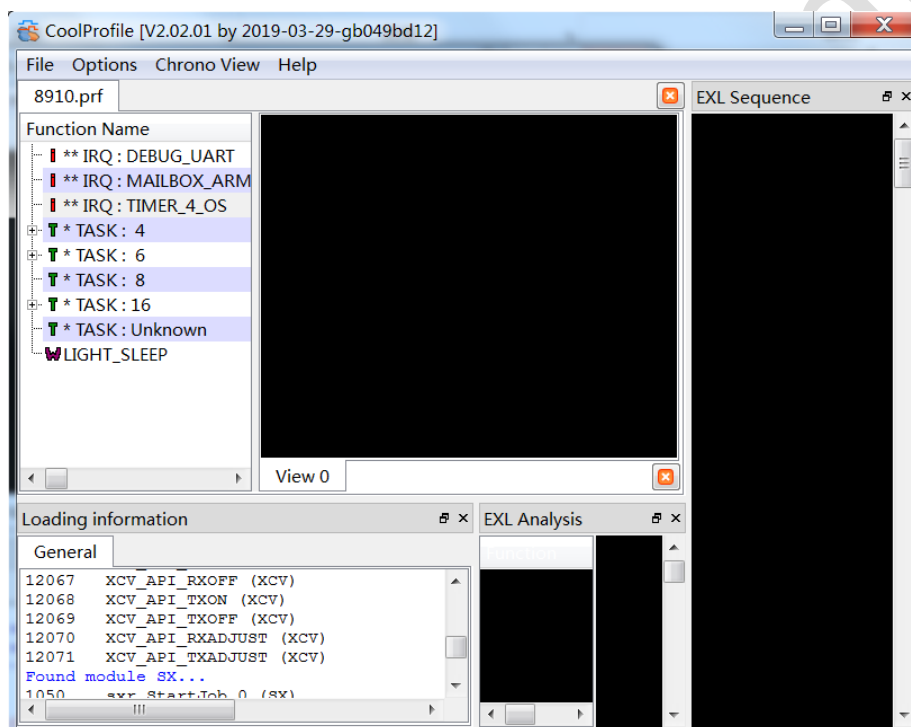


图 4-28 加载 profile 文件

即可分析 prf 文件。

4.7 离线分析

注意：8910 不推荐用该部分进行离线分析。

4.7.1 生成*.core(*.elf)

手机死机后，可以利用 elfdump 命令可以生成*.core(*.elf)文件，抓取*.core(*.elf)文件的

命令见章节 4.30 ，举例如下：

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

图 4-29 生成 elf 文件

4.7.2 建立离线分析环境

步骤如下：

1. 加载要分析的*.core (或*.elf)文件。

Coolwatcher 关闭的情况下，启动 coolhost.exe 程序，主界面如下：

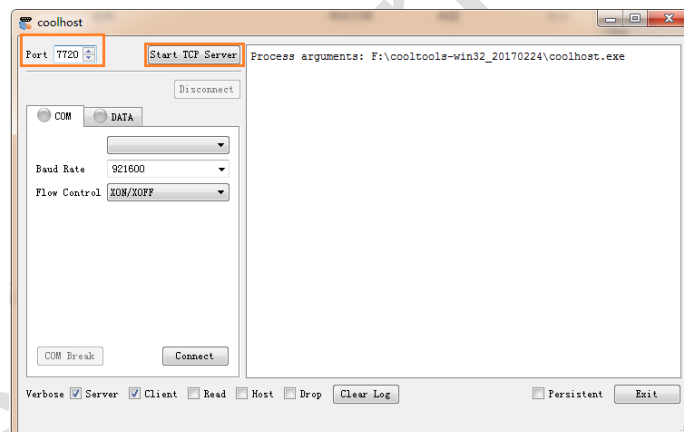


图 4-30 coolhost 主界面

- Port 端口号，可以随意定；
- 点击 Start TCP Server 按钮；
- 点开 DATA tab 页，右键点击 Add 菜单项，如下图，选择*.core (*.elf) 文件和 lod 文件，如图 4-31 所示。注意，先选择*.core (*.elf) 文件。

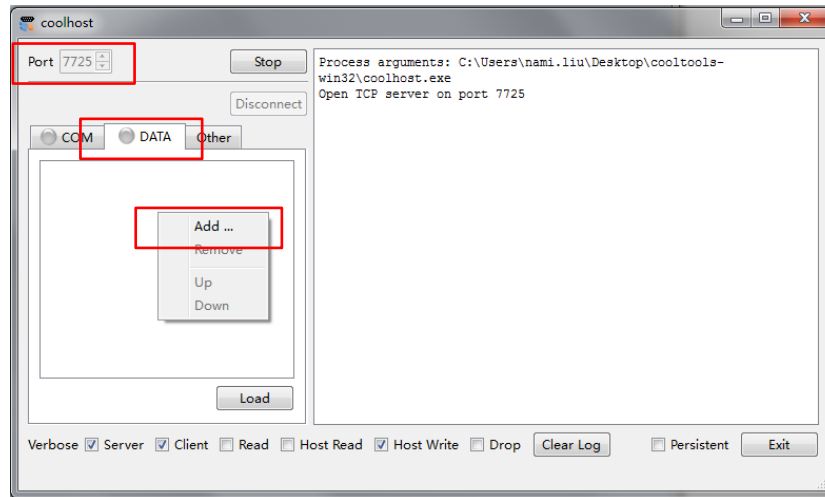


图 4-31 加载 elf 文件

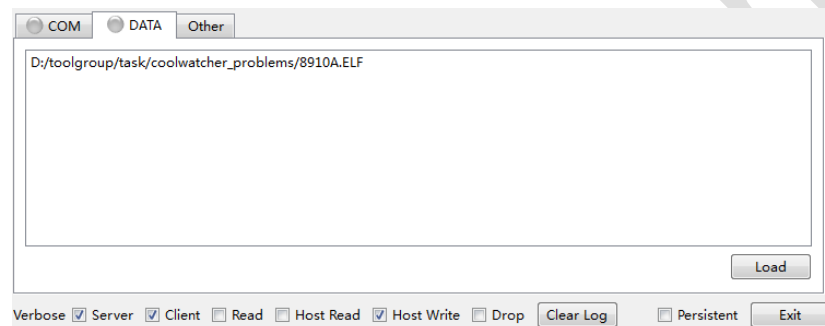


图 4-32 选择 elf 文件路径

- 加载文件：点击图 4-32 的 load 按钮，加载文件，效果如下图：

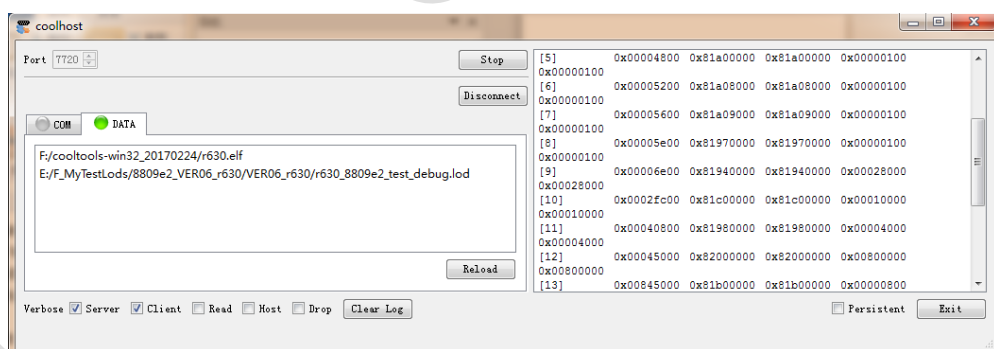


图 4-33 加载数据后界面显示

注意：该*.core (*.elf) 文件用 elfdump 命令生成，不是和 lod 一起发布的 elf 文件。

2. 启动离线分析环境

启动 coolwatcher，下图中的 lastcomport 项的数值要与图 4-31 中 Port 7725 的后两位

保持一致，如 Port 选择 7725，lastcomport 项的值为 25。

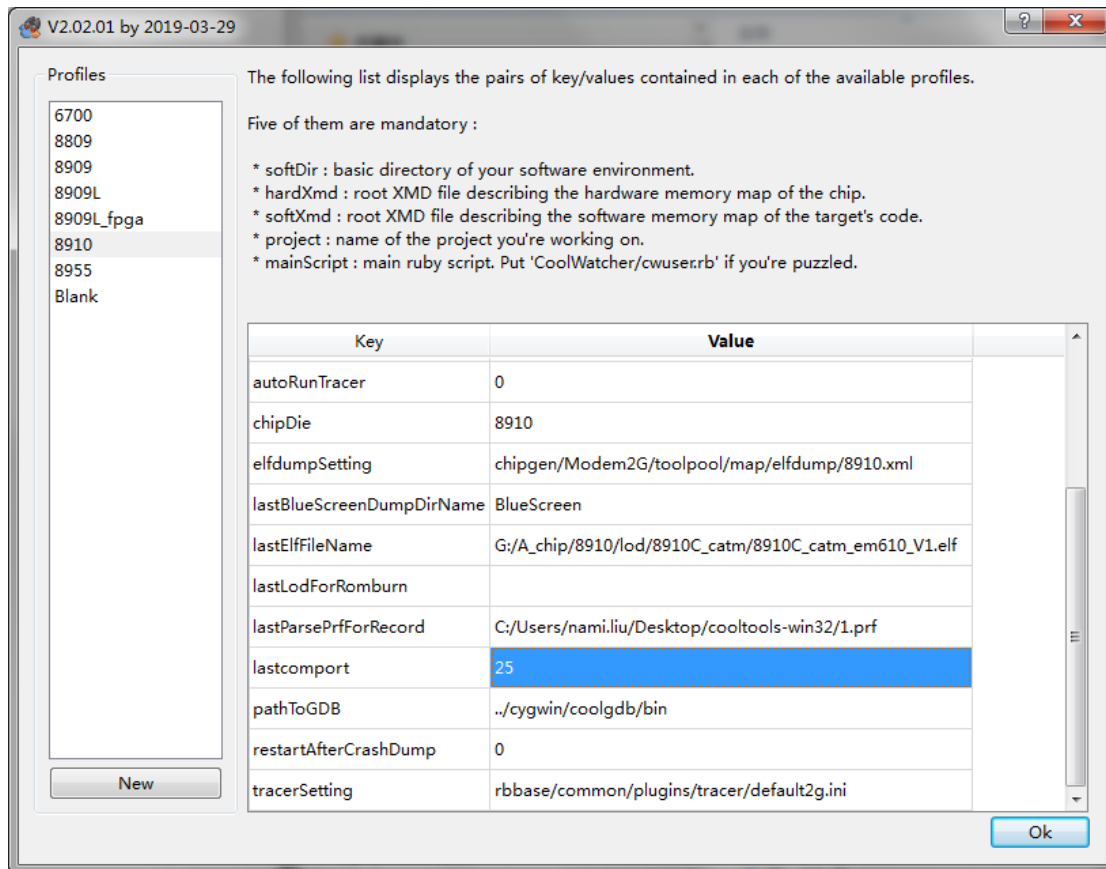


图 4-34 设置 lastcomport

启动 coolwatcher 后，即可进行离线分析。

4.7.3 离线分析 GDB

保持 4.7.2 环境不变，按照步骤 4.7.2 即可离线分析*.elf 文件。

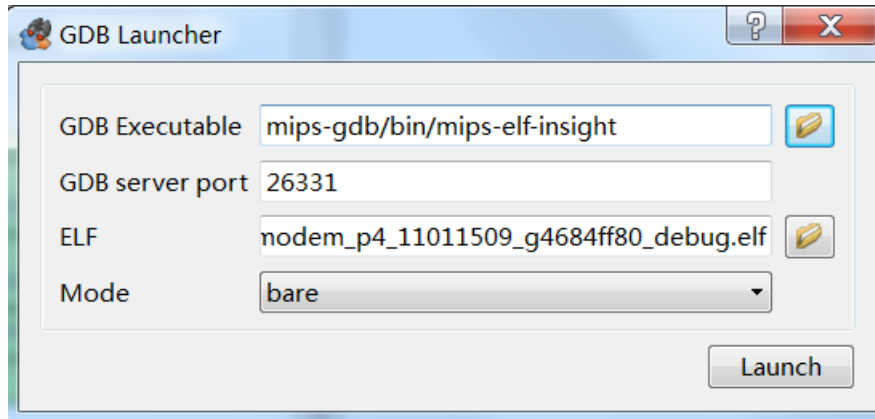


图 4-35 选择 elf 文件路径

注意：该配置框中选择的*.elf 文件，是与 lod 一起发布的*.elf 文件。

4.7.4 Elf Data Check

用于比较 dump 和原始 elf，检查代码是否有被改写。

启动过程

利用 4.7.2 章节介绍搭建离线环境后，点击 Tools->Elf Data Check 菜单，启动主界面图 4-37：

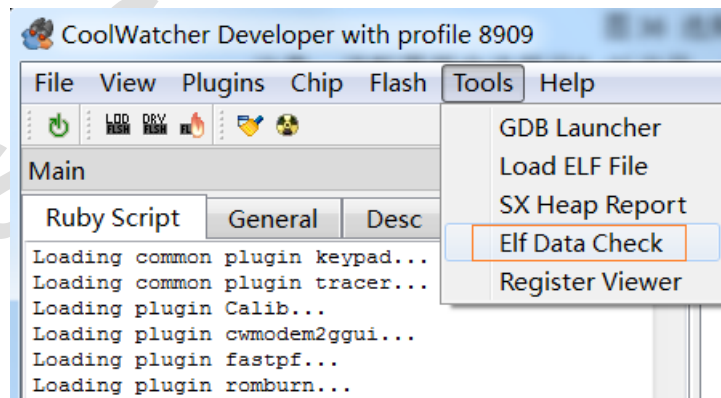


图 4-36 选择 Elf Data Check

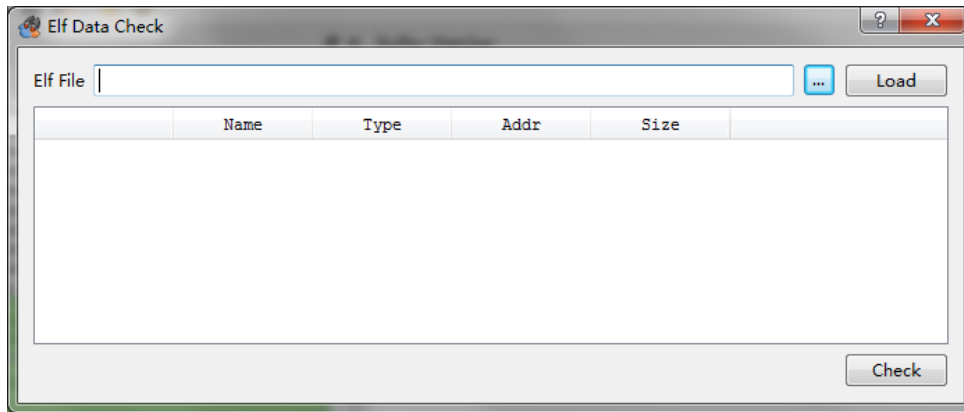


图 4-37 Elf Data Check 主界面

操作步骤：

1. 点击上图中的“load”按钮，选择与 lod 一起发布的*.elf 文件，点击“Check”按钮后如下图:

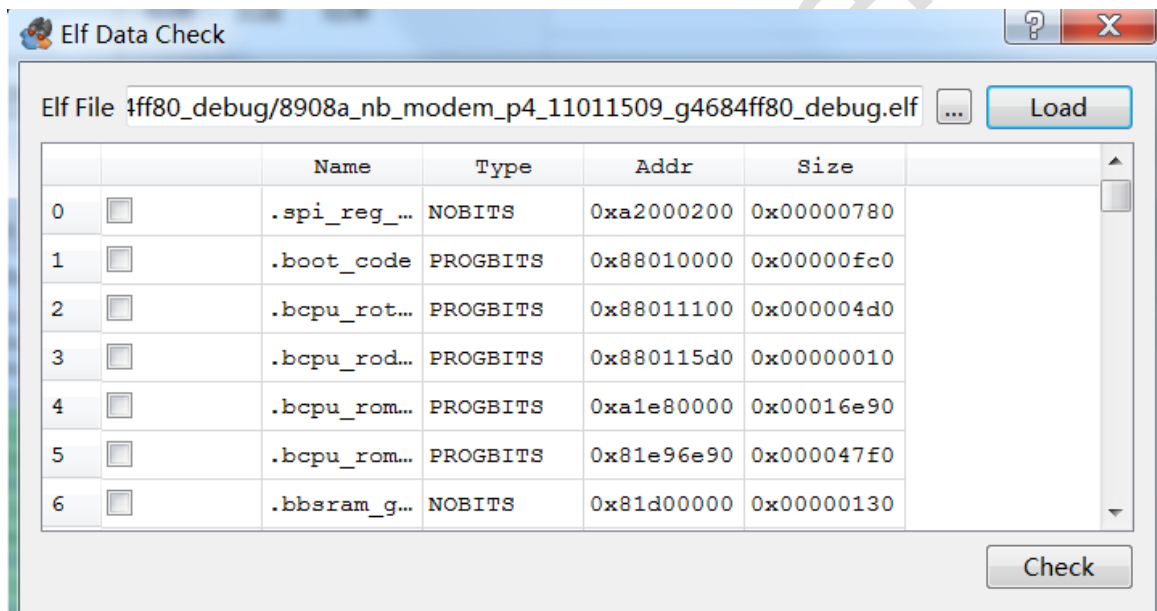


图 4-38 load elf 文件后界面显示

2. 选择待比较的项

通过操作首列 check box，选择待比较的项：

5	<input checked="" type="checkbox"/>	.bcpu_rom_ro...	PROGBITS	0x81e8e960	0x00002700
6	<input type="checkbox"/>	.bbsram	PROGBITS	0x81980000	0x00001400
7	<input type="checkbox"/>	.bbsramu	PROGBITS	0xa1981400	0x00000d40
8	<input checked="" type="checkbox"/>	.bbsram_glob...	PROGBITS	0x81983260	0x00000190

图 4-39 选择比较项

3. 比较和保存结果

点击图 4-39 中的 check 按钮，如果有不同，则会弹出下对话框：

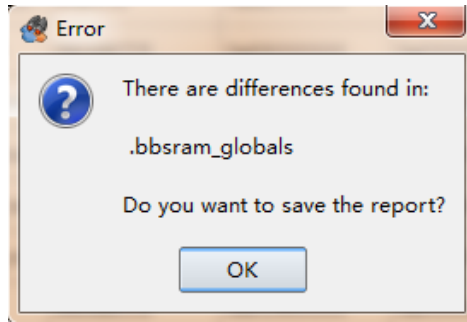


图 4-40 错误提示

点击 OK 按钮，选择/填写要保存的文件：

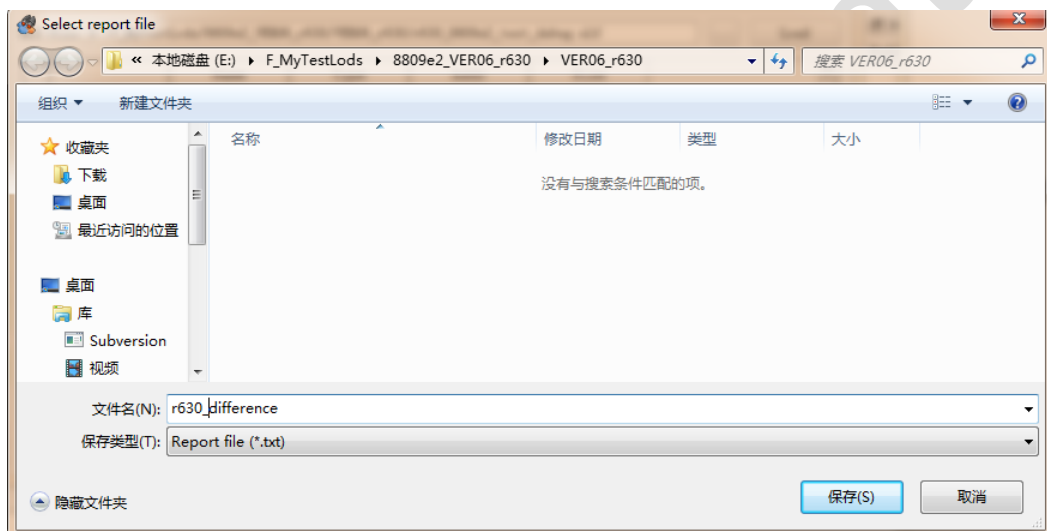


图 4-41 保存对比报告

不同处将会保存到文件中，效果如下图：

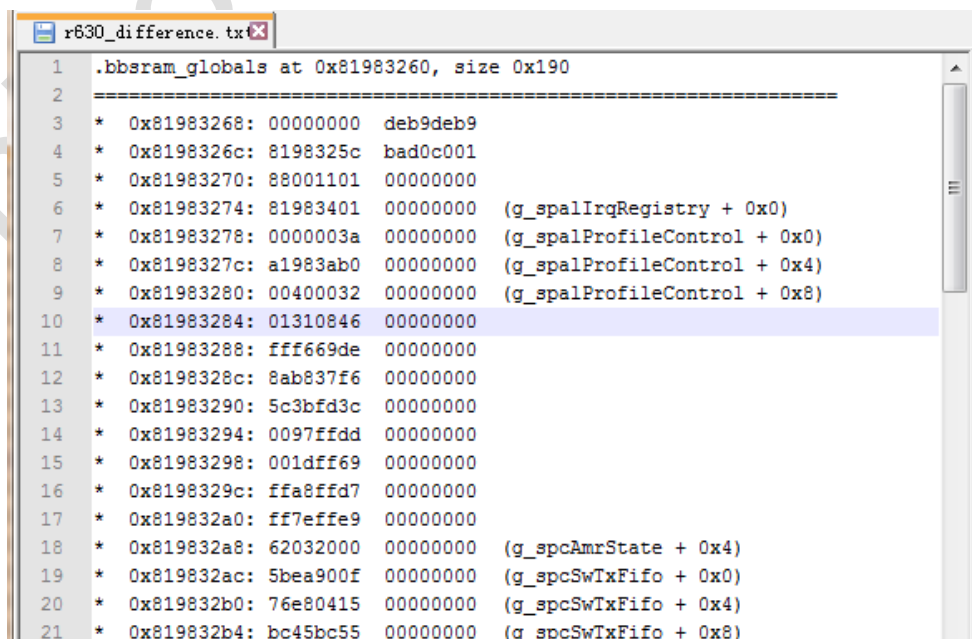


图 4-42 对比报告中不同处对比

4.8 Heap Report

4.8.1 SX Heap Report

点击主菜单 Tools->Heap Report 项，如下图，启动 Heap Report 设置对话框，如图 4-44。

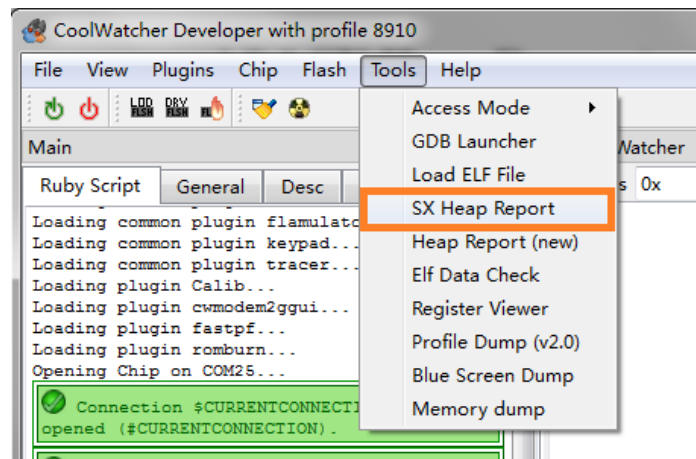


图 4-43 启动 Heap Report

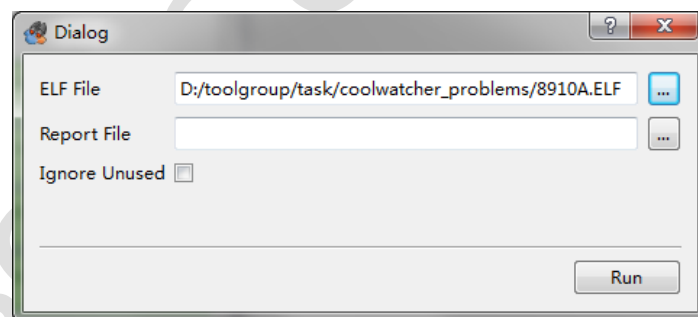


图 4-44 Heap Report 对话框

Elf 文件：与 lod 一起发布的 elf 文件；

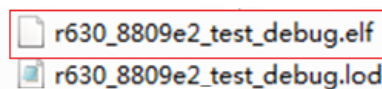


图 4-45 选择 elf 文件

Report File：要生成的文件。

选择或填写相关信息，点击 Run 按钮，生成 Report File，示例如下：

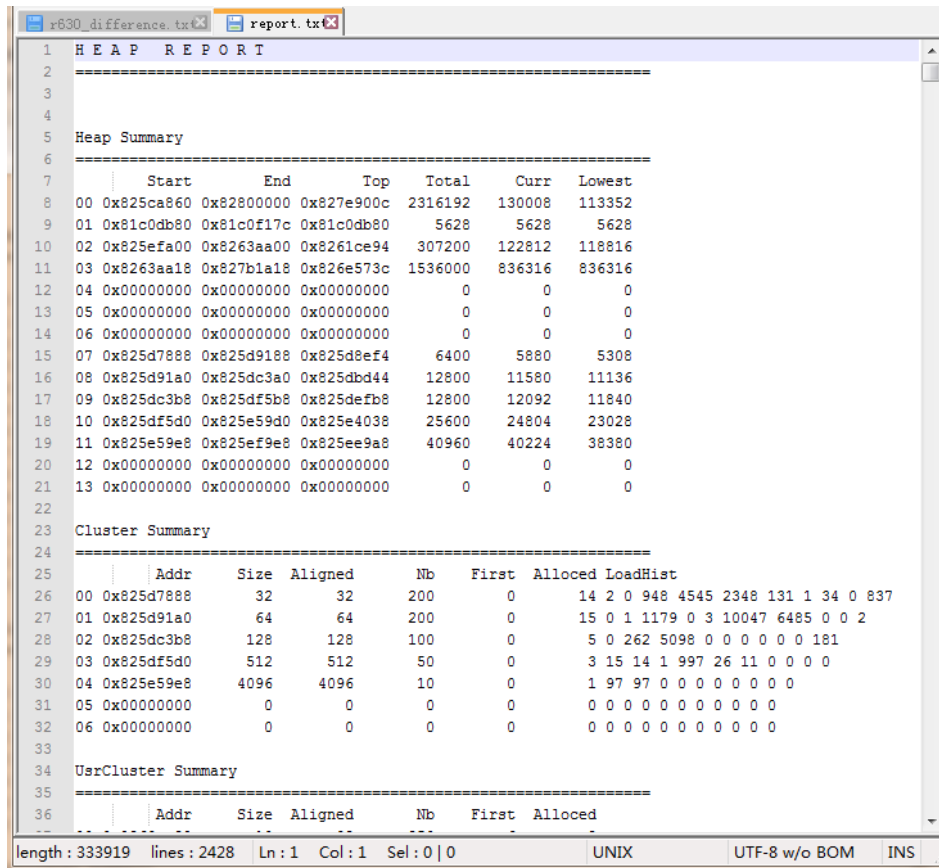


图 4-46 生成 Report File

4.8.2 Heap Report (new)

通过 Tools -> Heap Report (new)启动 Heap Report (new) , 如图 4-47。

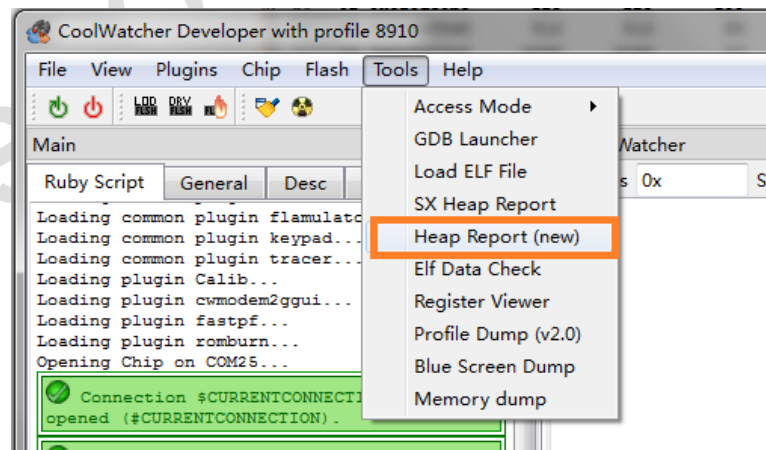


图 4-47 启动 Heap Report (new)

现在没有保留实际申请的 size , 显示的 size 包括系统管理开销 , 比实际申请的大。开销是 8 ~ 16

字节, 不固定。

申请块尾上有一个字节的 0xFD，可以用以检查越界。

4.9 Register Viewer

RegisterViewer 工具可以按 bit 位读取、编辑寄存器，同时具有查找功能。

点击主菜单 Tools->Register Viewer 项，如下图，启动 Register Viewer 页面，如图 4-49。

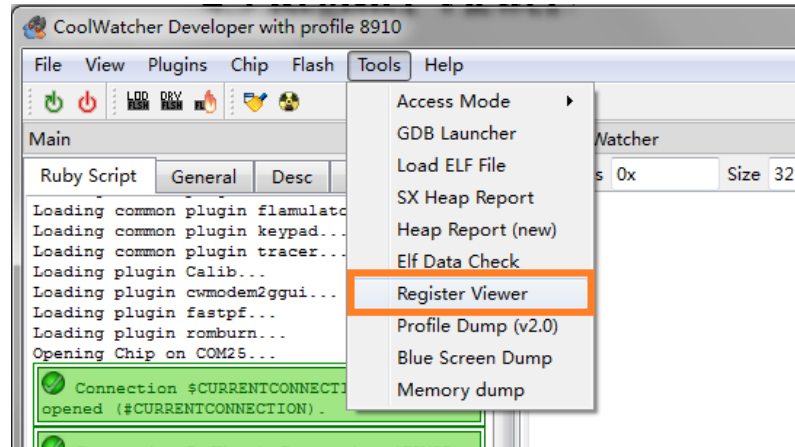


图 4-48 启动 Register Viewer

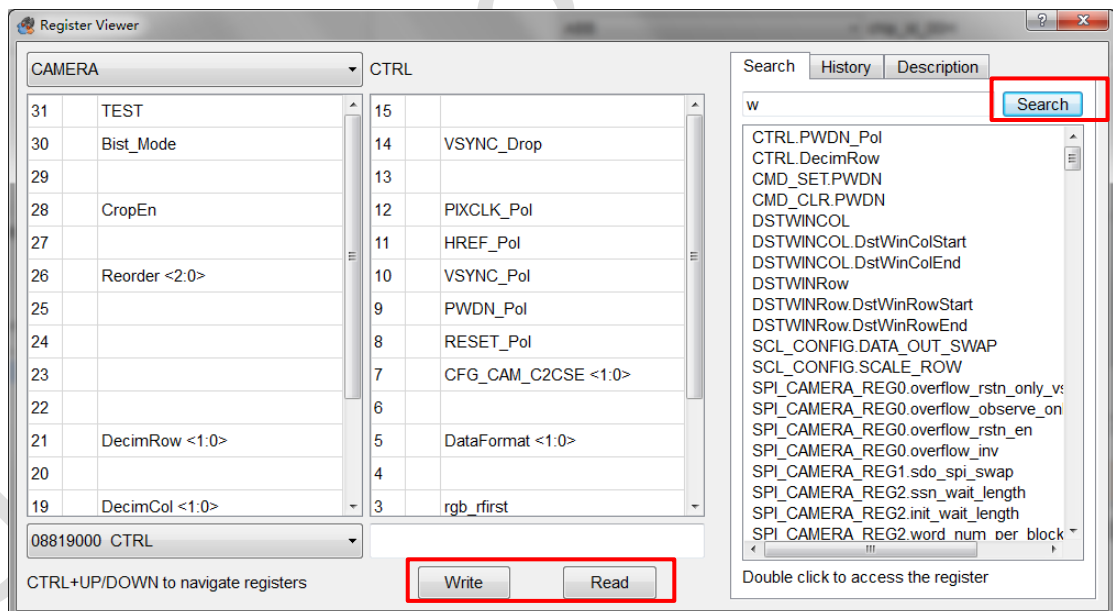


图 4-49 Register Viewer 主界面

4.10 Blue Screen Dump

4.10.1 启动 Blue Screen Dump

通过 tools→Blue Screen Dump 启动 Blue Screen Dump 主界面，如图 4-51 所示。

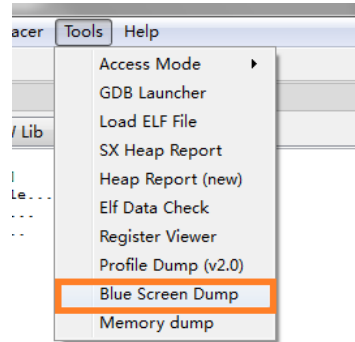


图 4-50 启动 Blue Screen Dump

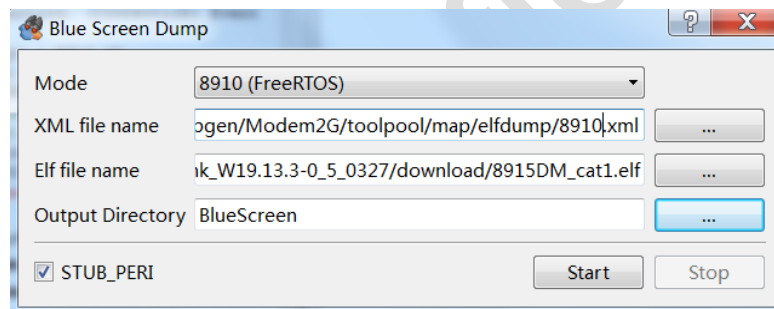


图 4-51 Blue Screen Dump 主界面

属性设置

Mode : ARM dump bin only、MIPS dump bin only、8910(FreeRTOS)、8909(FreeRTOS)、8955(FreeRTOS)五种模式，根据不同的芯片，选择不同的 mode。

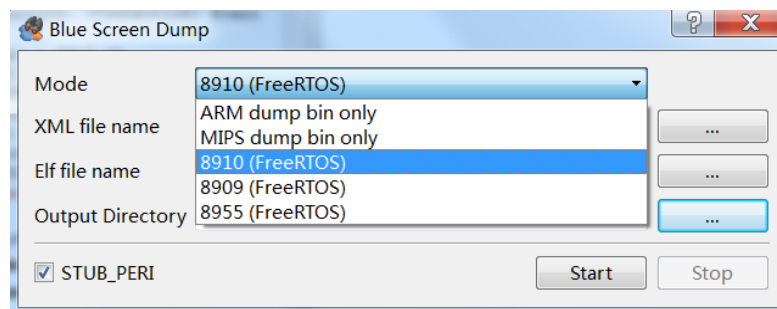


图 4-52 Blue Screen Dump 5 种 Mode

XML：用缺省的 elfdump xml 文件；

Output Directory：dump 文件输出目录，每次 dump 需选择新的目录；

STUB_PERI：默认 checked 状态 ☒ STUB_PERI，为快速 dump 模式 (software stub,with ADI/ISPI/RFSPI)。选中该模式时，工具会优先设置快速模式，如果设置失败，则用原有的 dump 模式。

注意：

1. 如果 dump 失败，在 command 窗口输入 'r 0' 测试下读取功能，如果读取失败，需先执行 'creconnect' 命令，再执行 'r 0'。读取正常的情况下，如果 STUB_PERI 模式仍然失败，则需把 ☒ STUB_PERI 改为 unchecked 状态 ☐ STUB_PERI。
2. Dump 结束后，窗口显示 "None of blue screen data has been dumped out!" 信息时，说明导出全部失败。

Log 信息

蓝屏导出过程的 log 信息会保存在与 dump *.bin 文件相同的文件夹中，命名格式为 dump_*.log，内容示例如下。

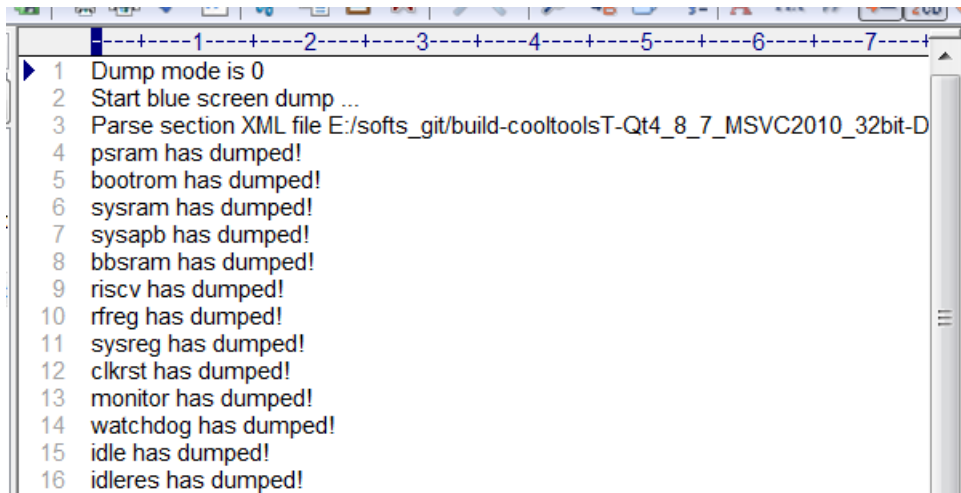


图 4-53 Blue Screen Dump log 信息

4.10.2 蓝屏分析

1. 8910 蓝屏分析

- 分析 AP

把 AP 的 ELF 拷贝到 dump 目录下，重命名为 ap.elf；

把 ap.cmm 拖到 TRACE32 中即可；

- 分析 CP

把 CP 的 AXF 拷贝到 dump 目录下，重命名为 cp.axf（通常就是这个名字）；

把 cp.cmm 拖到 TRACE32 中即可；

2. Dump 中目前包含 PMIC 寄存器，如果有需要后续还会添加；
3. AP FreeRTOS 切换 task 目前还不工作；
4. Cmm 也可以通过 coolhost 离线加载；

Coolhost 加载 ap.cmm 或者 cp.cmm 是一样的；

Coolhost 离线加载后，coolwatcher 启 GDB 也可以，但推荐使用 TRACE32。

4.11 Profile Dump

通过 Tools -> Profile Dump (v2.0) , 启动 Profile Dump , 主界面如图 4-55 所示。

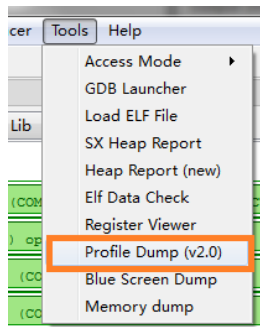


图 4-54 启动 Profile Dump (v2.0)

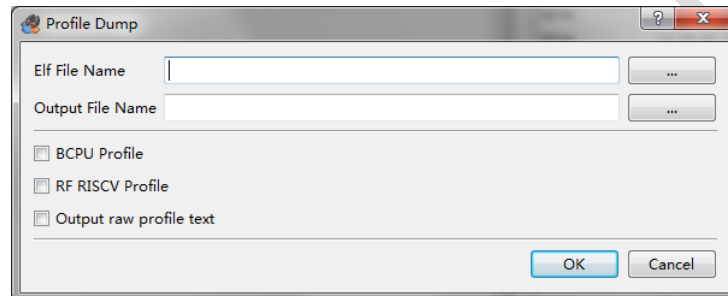


图 4-55 Profile Dump 主界面

通过该功能 dump 得到的 profile 文件可以通过 4.6.2 章节中提到的方法，使用 coolprofile.exe 工具进行分析。

Note :

用 coolprofile.exe 打开时，选择 8910.xcp。

4.12 Access Mode

Note : 谨慎使用本功能。

通过 Tools -> Access Mode 启动该功能。

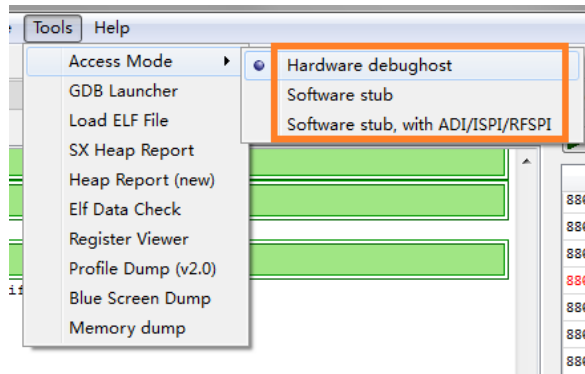


图 4-56 Access Mode 三种模式

现在 coolwatcher 读写有三种模式:

1. DebugHost 硬件；
2. 给 CPU 发命令，由 CPU 执行并返回；
3. 给 CPU 发命令，由 CPU 执行并返回，同时 CPU 处理特殊地址；
 - 8910：通过正常 ADI BUS 的操作方法读写 PMIC 寄存器；

建议只有读写 PMIC 时切换模式，不使用读写 PMIC 的功能时一定记得切回 Hardware DebugHost 模式。

软件读写 PMIC 会使用正常的 spinlock，不会有冲突。

4.13 芯片控制

工具提供了下列常用的对芯片进行控制的操作功能，大大提高了用户调试过程中对手机硬件控制的能力。

4.13.1 关闭芯片

点击工具条按钮 ，可以控制 UIS8910DM 关机。

4.13.2 重新启动芯片

点击工具条按钮，可以控制手机或开发板重新启动，便于调试用户程序。

4.13.3 芯片强制死机

点击菜单项 “Chip -> Force panic”，可以强制芯片死机，如图 4-58。

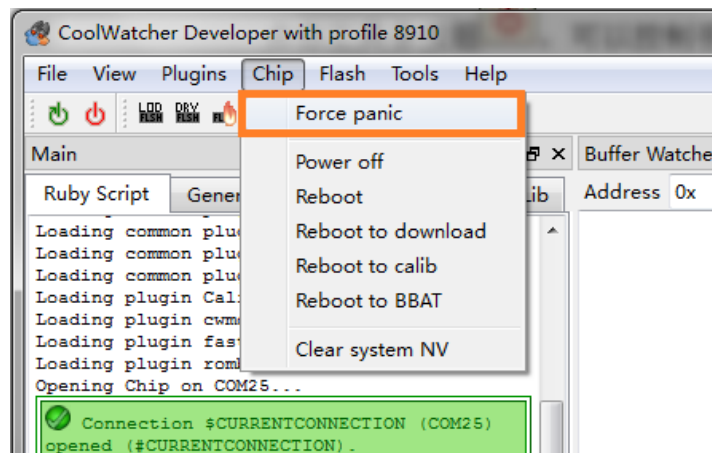


图 4-57 芯片强制死机

Force panic 是通过 DebugUart 给 CPU 发命令，如果 CPU 不能处理 DebugUart 命令（例如在高优先级 task 里面死循环），则无法触发 CPU 死机。

4.13.4 其他 chip 操作

图 4-54 中所示，除去芯片强制死机外，还涉及到其他芯片操作，介绍如下：

- Chip -> Power off：关机；
- Chip -> Reboot：重启；
- Chip -> Reboot to download：重启进入 ROM 强制下载；
 - 这个不是取代 FORCE DOWNLOAD PIN，而是提供更方便的手段，可靠性不如 PIN。

正规进入下载模式的方法是 FORCE DOWNLOAD PIN；

- Chip -> Reboot to calib：重启进入校准模式；

- 正规进入校准模式的方法是 SIMBA ENTER MODE ;
- Chip -> Reboot to BBAT : 重启进入 BBAT 模式 ;
- 正规进入 BBAT 模式的方法是 BBAT ENTER MODE ;

4.14 8910 USB 用法及注意事项

1 . 安装 8910 USB 驱动程序

安装 DTKWin\drivers\8910usb 文件夹中的驱动, 建议安装前先连接 PC 和 8910 模块。驱动安装完成后, 可在设备管理器中看到 8 个 USB 端口。

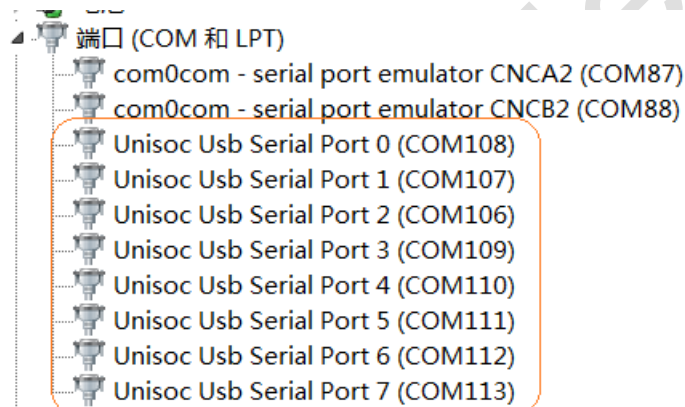


图 4-58 8910 USB 端口

2 . 用串口工具发送 AT 指令

Port0 为 AT 口, 用串口工具发送 AT 指令 `AT^TRACECTRL=0,1,2` 。

3 . 启动 coolwatcher

端口: 选择 USB Port4 对应的端口, 图 4-58 对应 COM100 ;

波特率: 没有要求, 默认即可。

流控: 务必选择 NONE ; 使用硬流控 hardware 和软流控 xon/xoff 也有 trace 输出, 不过 trace 可能会出问题。

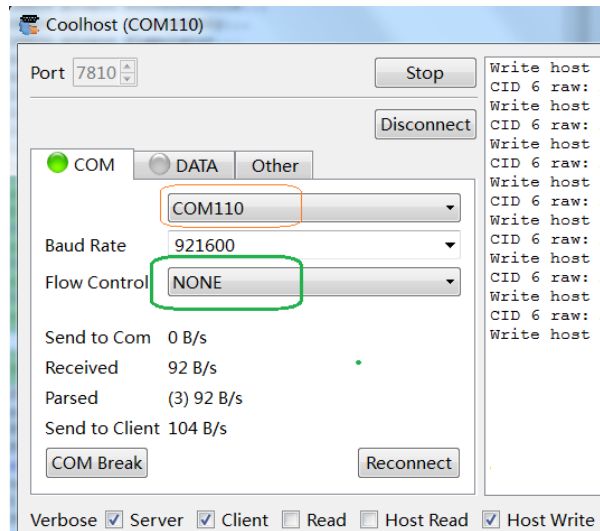


图 4-59 8910 USB 端口设置示例

如 coolwatcher 用于 itest 系统，流控、波特率、串口可通过 profiles\8910.cwp 配置。

4. 其他注意事项

1) 寄存器读取

不能读取 INT_REG_DBG_HOST 寄存器。

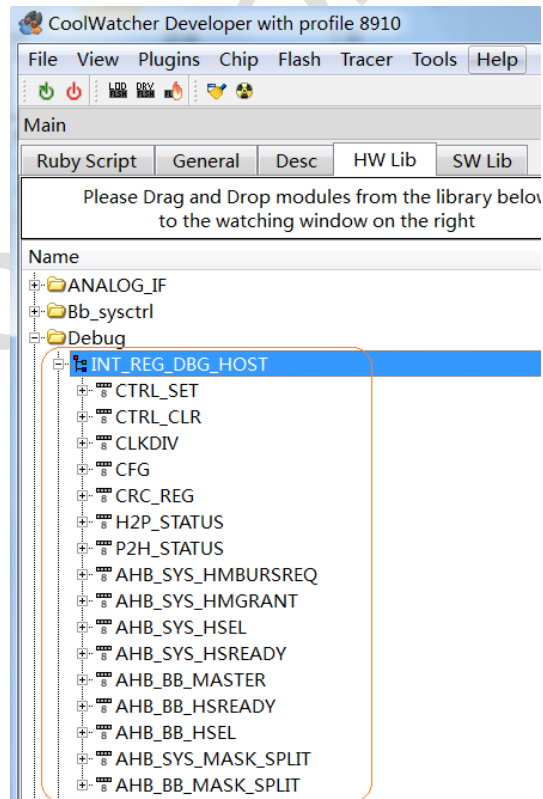


图 4-60 8910 INT_REG_DBG_HOST 寄存器示意图

2) USB 蓝屏导出功能 暂时不能用

4.15 串口二进制数据中的 trace 的回放

此功能仅适用的场景：因条件限制不能用 coolwatcher 抓取、保存 trace。借用其他工具把从 UART 口输出的数据保存为二进制文件，并且需要用 coolwatcher 分析二进制数据中的 trace。

串口数据中的 trace 不包含完整的时间信息，如果分析问题时需查看某两条 trace 的(大概)时间差，仅仅利用保存的二进制数据是不够的。

为此提供下面解决方案：

1. 抓包工具(需自行修改)

工具保存串口数据时，需增加时间戳信息，增加规则如下：

- ✚ 创建 LOG 文件时，自动插入一个 timestamp 的 HOST 包；
- ✚ 如果 UART 一段时间没有数据（例如 1 秒），自动再插入一个 timestamp 的 HOST 包；
- ✚ 如果 UART 长时间没有数据，每秒插入一个 timestamp 的 HOST 包；
- ✚ 支持上自动分段的功能，避免极大的二进制文件，单个文件不超过 10M；

2. Coolwatcher

coolwatcher 导入 bin 文件时，会处理时间戳 HOST 包，使用时间戳 HOST 包的时间，在下一个 timestamp 到达之前用 tick 累加。

3. 时间戳 HOST 包格式

包结构如下：

```
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: 均用默认值，且不能修改；
- ⊕ date：利用公式 “year << 16 + month << 8 + day” 计算 date。year、month、day 分别代表年月日；
- ⊕ ms：毫秒 从 0 点至当前时间的总毫秒数；
- ⊕ 大小端：时间戳 HOST 包用小端方式保存。

下图为时间戳在 coolwatcher tracer 显示窗口中的效果。

759	10:49:43.275			DateTime: 2020-04-24 10:49:43.275
760	10:49:43.275	3988		osi_init
761	10:49:43.518	4293	L1S 01	Start near cell scheduler.\n
762	10:49:43.536	4296	/I	g_CswHeapSize 0x64000\n
763	10:49:43.536	4296	/I	g_CosHeapSize 0x82000\n
764	10:49:43.536	4484	/T	\nRAI_TaskInit Start \n

Raw	Description
ad 00 08 a2 18 04 e4 07 6b d6 52 02k.R.

图 4-61 时间戳信息显示示意图

4.16 命令行操作

本工具提供了一些开发中常用的操作命令，用户可以很方便的在命令行输入框输入这些命令，完成相应的操作，并将操作结果显示在 “Ruby script” 框内。

4.16.1 端口操作

常用的端口操作命令如下列表：

5.

命令	参数	示例	备注
copen	(NUM,BR=BR_AUTOMATIC)	copen(2,115200)	打开 COM2
reop		reop	重新打开当前端口

4.15.2 Flash 编程

常用的 Flash 编程命令如下列表：

命令	参数	示例	备注
fastpf	(flash_programmer_filename, lod_filename, disable_event_sniffer = true)	fastpf("c:\xxxx_ramrun.lod " , " c:\\lod")	烧写 Flash
fastSectorEraser	(flash_programmer_filename, sector_list, disable_event_sniffer = true)	fastSectorEraser("c:\xxxx_r amrun.lod" , " [0x01000000,0x01200000] ")	擦除扇区

4.15.3 读 Flash

常用的读 Flash 操作命令如下列表

命令	参数	示例	备注
r	(addr)	r x01004000	32 位写
r32	(addr)	r32 0x01004000	32 位写

r16	(addr)	r16 0x01004000	16 位写
r8	(addr)	r8 0x01004000	8 位写

4.15.4 写 Flash

常用的写 Flash 操作命令如下列表

命令	参数	示例	备注
w	(addr,val)	w(0x01004000,0xffffffff)	32 位读
w32	(addr,val)	W32(0x01004000,0xffffffff)	32 位读
w16	(addr,val)	W16(0x01004000,0xffff)	16 位读
w8	(addr,val)	W8(0x01004000,0xff)	8 位读


4.15.5 其他命令

命令	参数	示例
elfdump	elf 文件 [,xml 文件名] xml 文件名可以不写, 如果写需要给出绝对路径	elfdump "test.elf" 或者 elfdump "test.elf" , "D:/8809.xml"
dump	(str_filename,address,nbwords) 读取一段连续的 flash 区域到指定文件。	Dump("c:\\xxxx.lod" ,0x01004,0x100)
chipID	无参数, 直接返回芯片 ID	000,0x1000)
flashReadStatus	无参数	

flashSectorErase	flashSectorErase(addr)	
flashBlockErase	flashBlockErase(addr)	
flashBlock32kErase	flashBlock32kErase(addr)	
flashChipErase	flashChipErase()	
xcvRead	xcvRead(addr)	
xcvWrite	xcvWrite(addr,data)	

4.17 其他功能


4.17.1 Kill 当前运行的程序

点击菜单项 “File->Kill command thread” 或工具条按钮 ，可以强制终止当前正在运行的程序（一般为用户在命令行输入的命令线程）。

4.17.2 Kill 所有运行的程序

点击菜单项 “File->Kill all threadd”，可以强制终止正在运行的全部（一般为用户在命令行输入的命令线程）。

4.17.3 清除脚本输出信息

点击菜单项 “View->Clear Screen” 或工具栏按钮 ，可以清空左下方的 “Ruby script” 区域的输出信息。

4.18 Linux 串口配置

在应用程序 coolwatcher.exe 所在文件夹中建立子文件夹 comport。

为/dev/ttyUSB0、/dev/ttyUSB1 创建符号连接文件

如 `ln -s /dev/ttyUSB0 comport/COM1`

`ln -s /dev/ttyUSB1 comport/COM2`

访问端口。

5 Q & A

6 附录 Appendix

无