



SIM85XX系列_Log抓取_应用文档

智能模组

芯讯通无线科技(上海)有限公司

上海市长宁区临虹路289号3号楼芯讯通总部大楼

电话: 86-21-31575100

技术支持邮箱: support@simcom.com

官网: www.simcom.com

名称:	SIM85XX系列_Log抓取_应用文档
版本:	1.00
日期:	2022.3.14
状态:	已发布

版权声明

本手册包含芯讯通无线科技（上海）有限公司（简称：芯讯通）的技术信息。除非经芯讯通书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播，违反者将被追究法律责任。对技术信息涉及的专利、实用新型或者外观设计等知识产权，芯讯通保留一切权利。芯讯通有权在不通知的情况下随时更新本手册的具体内容。

本手册版权属于芯讯通，任何人未经我公司书面同意进行复制、引用或者修改本手册都将承担法律责任。

芯讯通无线科技(上海)有限公司

上海市长宁区临虹路289号3号楼芯讯通总部大楼

电话：86-21-31575100

邮箱：simcom@simcom.com

官网：www.simcom.com

了解更多资料，请点击以下链接：

<http://cn.simcom.com/download/list-230-cn.html>

技术支持，请点击以下链接：

<http://cn.simcom.com/ask/index-cn.html> 或发送邮件至 support@simcom.com

版权所有 © 芯讯通无线科技(上海)有限公司 2022，保留一切权利。

关于文档

版本历史

版本	日期	作者	备注
1.00	2022.3.14	宫安娜	第一版

适用范围

目录

关于文档.....	3
版本历史.....	3
适用范围.....	3
1 介绍.....	5
1.1 本文目的.....	5
1.2 参考文档.....	5
1.3 术语和缩写.....	5
2 使用 adb 抓取 log.....	6
2.1 下载 adb 工具包.....	6
2.2 打开 adb 工具。.....	6
2.3 查看已连接的设备(adb devices).....	6
2.4 输入抓取命令.....	6
2.5 其他抓取方式.....	8
2.5.1 通过串口抓取 putty.....	8
2.5.2 通过 wifi 抓取.....	8
2.5.3 离线 log.....	8
3 Ylog 抓取.....	8
3.1 概述.....	8
3.2 环境配置.....	9
3.2.1 测试资源准备.....	9

3.2.2 PC 端配置	9
3.2.3 手机端配置.....	9
3.3 YLog 使用说明.....	9
3.3.1 YLog 设置.....	10
3.3.2 场景设置.....	10
3.4 YLog 抓取.....	14
3.5 直接拷贝.....	14
4 通过串口工具 putty 抓取 log.....	15
5 不同场景下抓取 log 的方式.....	17
5.1 可以使用 adb（app 闪退、报错；系统开机问题：开机慢 动画相关问题）	17
5.2 无法使用 adb（死机；冻屏；黑屏；没有 usb 口）	17
5.3 不需要使用 adb 工具.....	18
5.4 特殊情况（冻屏）	18
5.5 网络问题.....	18

1 介绍

1.1 本文目的

本文主要介绍 Log 的抓取方式流程。

参考此应用文档，开发者可以很快理解并快速针对不同场景抓取 Log。

1.2 参考文档

1.3 术语和缩写

2 使用 adb 抓取 log

2.1 下载 adb 工具包

adb 工具在 Android sdk 中的 platform-tools 文件夹可以找到

如果没有 Android sdk 可以在下面地址下载

FTPAddress: 117.131.85.140:9021

Account: SIMCom-cust1

Password: Abc123!

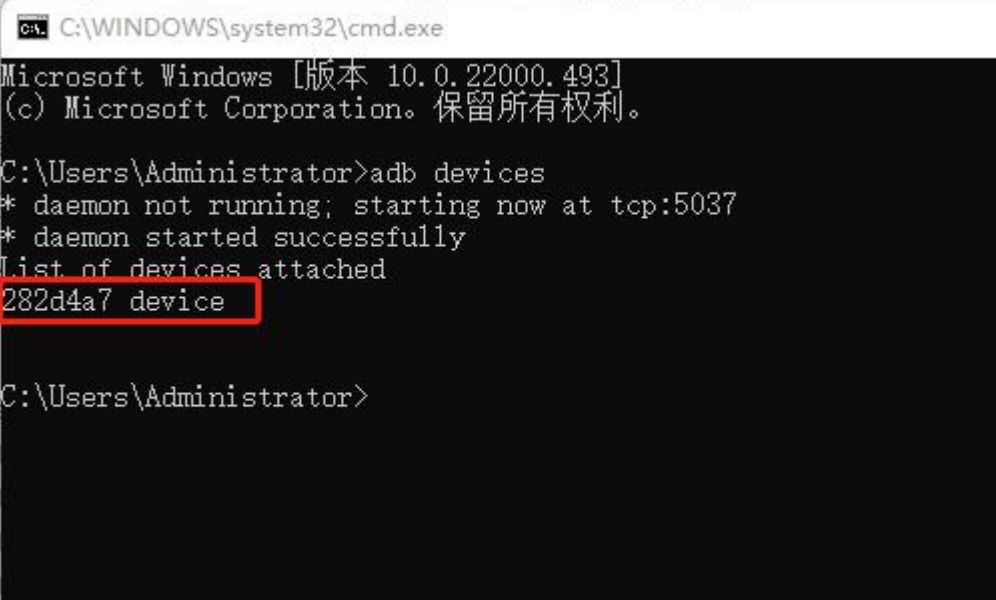
路径: \Tools\Adb&fastboot

2.2 打开 adb 工具。

打开 windows 系统 cmd 工具 定位到 adb.exe 所在的文件夹 输入命令即可

2.3 查看已连接的设备 (adb devices)

adb devices,如果按了回车后出现一段字母数字组合就说明设备连接成功了!



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.22000.493]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\Administrator>adb devices
* daemon not running; starting now at tcp:5037
* daemon started successfully
List of devices attached
282d4a7 device

C:\Users\Administrator>
```

如果没有设备显示 请检查设备开发者选项中 USB 调试是否开启 USB 线是否连接

2.4 输入抓取命令

adb logcat >1.txt 这时会将所有 log 保存至 1.txt 中
复现问题, 出现问题后按下 Ctrl+C 来取消抓取日志

adb logcat -c 清空旧日志

以下示例中，1.txt 文件在 ‘C:\Users\Administrator>’ 路径下

```
C:\Users\Administrator>adb logcat >1.txt
C:\Users\Administrator>
```

注意：如果出现这个日志，是因为最终的 logcat 进程退出，而退出的原因是 log buffer size 设置过小导致，而默认 size 为 256KB，如果你的程序长时间运行，并且产生了大量的日志，最终日志缓存的大小肯定是超过了默认的 256kb

```
C:\Users\Administrator>adb logcat >1.txt
logcat: Unexpected EOF!
```

This means that either the device shut down, logd crashed, or this instance of logcat was unable to read log messages as quickly as they were being produced.

If you have enabled significant logging, look into using the -G option to increase log buffer sizes.

这时需要打开设置--关于手机--连续按版本号 7 次进入开发者模式
然后进入系统--开发者选项--日志记录器缓冲区大小
选择合适大小即可



Kernel log 抓取命令

```
adb shell dmesg > 1.txt
```

也可以使用批处理文件抓取所有的日志 方便开发人员分析

```
mkdir Android_Log
```

```
start cmd /c "adb logcat -b radio -v time>Android_Log\radio.txt"
```

```
start cmd /c "adb logcat -b events -v time>Android_Log\events.txt"
start cmd /c "adb logcat -b system -v time>Android_Log\system.txt"
start cmd /c "adb shell top>Android_Log\top.txt"
start cmd /c "adb shell dmesg>Android_Log\kernal.txt"
adb logcat -v time>Android_Log\main.txt
@pause
这个批处理文件在下载的 adb 工具里面
```

2.5 其他抓取方式

2.5.1 通过串口抓取 putty

串口中输入 logcat

2.5.2 通过 wifi 抓取

当需要调试 usb 功能 usb 被占用的时候 可以使用 wifi 抓取

连接 usb 线输入 adb tcpip 5555 让设备的 adb 切换到 tcp 模式

设备和电脑处于同一个局域网下

拔掉 usb 线输入 adb connect xxx.xx.xx.xxx:5555

2.5.3 离线 log

adb shell

nohup logcat > data/anr/1.txt &

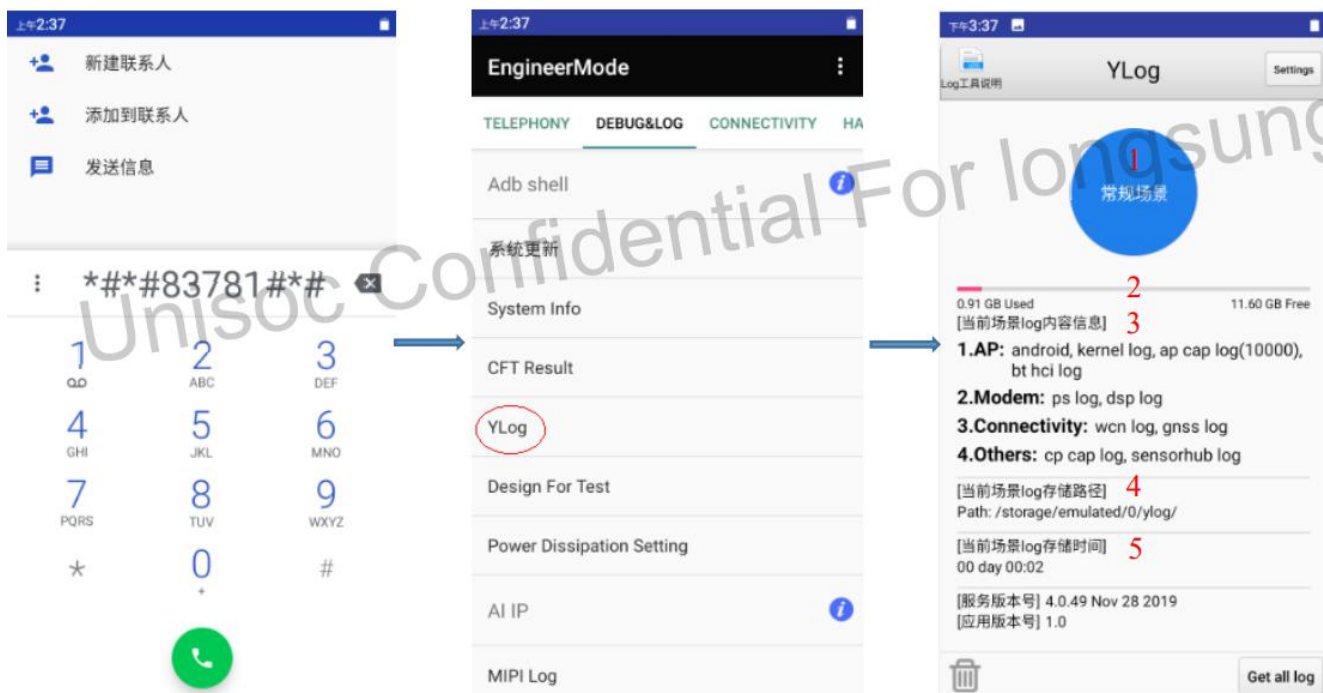
在 bug 复现之前输入，复现后导出即可

3 Ylog 抓取

3.1 概述

展锐 Log 抓取方式分为在线方式和离线方式。在线方式指 PC 方式抓取 Log。离线方式可离线工作并获取 Log，早期叫 SLog，后来升级为 YLog。YLog 是一个 C 语言编写的守护程序，运行于安卓系统的 native 层，用于收集手机中各模块输出的各种 Log，以文件形式保存在手机内存或手机 SD 卡中，便于开发者查看及定位分析问题。

YLog 通过工程暗码进入，在拨号键盘输入 `*##83781##` 进入工程模式，在菜单 **DEBUG&LOG** 中选择 YLog 选项，进入 YLog 功能主界面



1. YLog 开关及 YLog 当前场景：蓝色表示开启，灰色表示关闭。
2. SD 卡使用情况：红色表示已使用的空间，灰色表示剩余空间。
3. 当前场景 Log 内容信息：表示当前场景下 Log 输出的模块信息，如 AP Log, Modem Log 等，与选择的场景息息相关。
4. 当前场景 Log 存储路径：表示 Log 当前的储存位置。
5. 当前场景 Log 存储时间：表示 Log 从开始抓取的持续时间，停止抓取后计时清零。

3.2 环境配置

3.2.1 测试资源准备

若设备插有 SD 卡，YLog 默认储存在 SD 卡中；若设备不插 SD 卡，YLog 则默认储存在手机内存中。为确保 Log 信息被完整储存，测试前最好需准备一张 16G&10 速及以上的 SD 卡。如果测试时间较长，如 Monkey 测试等，建议选择存储空间更大的 SD 卡；否则 SD 卡存储容量太小或不插 SD 卡，会导致 Log 保存不完整，部分 Log 数据丢失。

3.2.2 PC 端配置

安装 adb 环境及驱动，手机端插入 USB 线后可在 PC 端正确识别端口。

下载批处理导出工具 Log4Android2PC。 安装 python 工具。

3.2.3 手机端配置

手机端已插入 SD 卡。

3.3 YLog 使用说明

YLog 的使用分为设置/抓取/储存/导出/回放五个阶段。

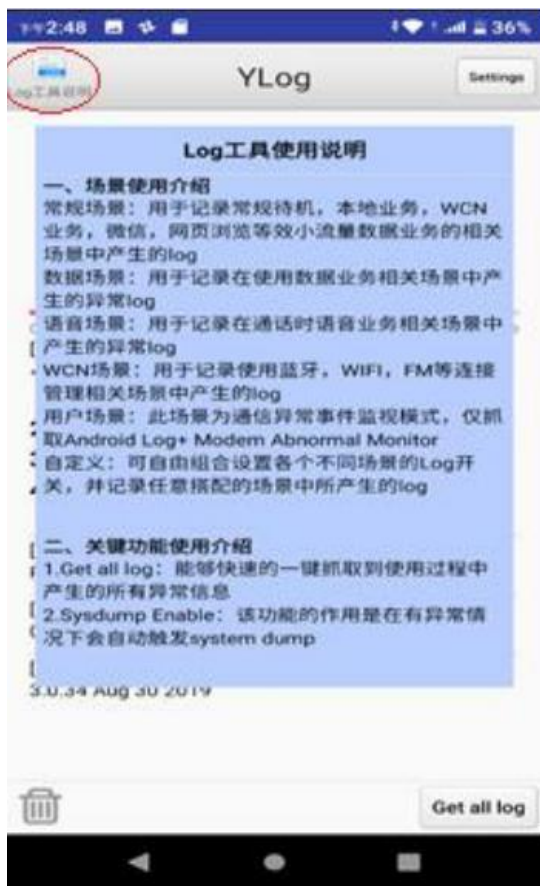
3.3.1 YLog 设置

YLog 设置分为“场景选择”和“高级设置”，YLog 界面右上角“Settings”选项，可以对 YLog 做相关 设置



3.3.2 场景设置

场景使用介绍见 YLog 界面左上角“Log 工具说明”



3.3.2.1 常规场景 用于记录常规待机，本地业务，WCN 业务，微信，网页浏览等效小流量数据业务的相关场景中产生的 log。YLog 默认为“常规场景”。

3.3.2.2 数据场景 用于记录在使用数据业务相关场景中产生的异常 log。在有独立 AGDSP 的芯片上，数据场景与常规场景的区别在不输出 AGDSP Log。在无独立 AGDSP 的芯片上，数据场景与常规场景相比输出的 Log 类型一致。

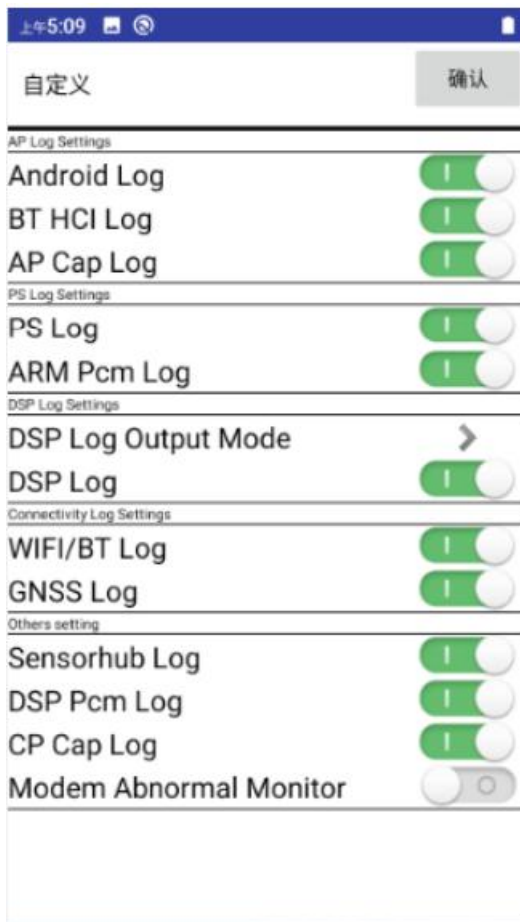
3.3.2.3 语音场景 用于记录在通话时语音业务相关场景中产生的异常 log。Unisoc Confidential For longsung 与常规场景相比，多输出 arm pcm log、dsp pcm log

3.3.2.4 WCN 场景 用于记录使用蓝牙，Wi-Fi，FM 等连接管理相关场景中产生的 log。在有独立 ADSP 的芯片上，WCN 场景与常规场景的区别在不输出 ADSP Log。在无独立 ADSP 的芯片上，WCN 场景与常规场景相比输出的 Log 类型一致。

3.3.2.5 用户场景 此场景为通信异常事件监视模式，仅抓取 Android Log+ Modem Abnormal Monitor



3.3.2.6 自定义场景 可自由组合设置各个不同场景的 Log 开关，并记录任意搭配的场景中所产生的 log。输入框内可改自定义场景的名称，确定后可进入编辑界面



AP Log Settings: AP 侧的 Log 包括 Android Log、BT HCI Log 和 AP Cap Log。其中 Android log 开关打开后，生成的 Log 包括 main Log、system Log、radio Log、event Log、kernel Log 和 crash Log

PS Log Settings: 可以开关 PS Log 和 ARM Pcm Log。其中 ARM Pcm Log 可以实现 WCN Log 和语音 Log 同时抓取，在抓取 ARM Pcm Log 同时，需打开 PS Log 开关

DSP Log Settings: 可以开关 DSP Log，同时可以设置 DSP Log 输出模式，可以选择 Output from UART 或者 Output with Modem Log

Connectivity Log Settings: 可以开关 WIFI/BT Log 和 GNSS Log

Others Settings: 可以开关 Sensorhub Log, DSP Pcm Log 和 Cap Log。其中 Dsp Pcm Log 可以实现 WCN Log 和语音 Log 同时抓取，在抓取 Dsp Pcm Log 时，需打开 DSP Log 开关。分析通话音质问题时需要抓取 Dsp Pcm Log

Modem Abnormal Monitor: Modem 异常监控开关。与 PS 以及 DSP LOG 开关互斥。开启异常监控开关后，PS LOG 和 DSP LOG 开关会自动关闭；开启 PS LOG/DSP LOG 开关后，异常监控开关会自动关闭。启动“Modem 异常监控”选项，即在不存 MODEM log 且不使用 PC 抓 log 时，启动通信异常事件功能

说明: 当从其它场景切换为自定义场景选择 Log 开关时，会默认将前一场景的 Log 开关选项全部勾选。

3.4 YLog 抓取

UserDebug 版本 YLog 抓取 UserDebug 版本 YLog 开关默认为蓝色开启状态。只需确保 YLog 正常抓取即可，即 Log 大小逐渐变大，存储时间持续变长。

User 版本 YLog 抓取

步骤 1 User 版本 YLog 开关默认为灰色关闭状态，故需手动开启。开启后弹出提示信息“保持 Log 开启状态重启手机，以确保 Android Log 所有功能生效”，是因为引入了 Log 主动丢失策略，以便关键 Log 能及时打印；如需要在 User 版本抓取全部 Log，则需要根据提示信息重启手机。

步骤 2 确保 YLog 正常抓取，Log 大小逐渐变大，存储时间持续变长。

YLog 存储 在储存介质的根目录下，会生成一个以 Ylog 命名的文件夹。若设备插有 SD 卡，YLog 文件夹会在外部存储 SD 卡的根目录下生成。若设备无 SD 卡，YLog 文件夹会在设备内部存储的根目录下生成。

4.4 YLog 导出 YLog 导出方式有两种，直接拷贝和工具导出。

3.5 直接拷贝

步骤 1 进行 get all log 操作。get all log，即一键抓取功能，快速抓取使用过程中产生的所有异常信息，可直接从 SD 卡中拷贝出完整目录结构的 Log 文件夹，去除对 Log4Android2PC 工具的依赖性，需插入 SD 卡。执行 get all log 后将自动生成 phonedata 和 logbuffer：Phonedata - 将以前 Log4Android2PC 功能在手机端实现，抓 Log 不依赖 PC 导出工具，使用更方便。

logbuffer - 导出 android 自身机制 buffer 的 Log，同时导出 bugreport 有助于分析无 log 和功耗相关问题。

步骤 2 手机端插入 USB 线连接 PC，选择 MTP 模式



步骤 3 在 PC 端打开 SD 卡，找到 YLog 文件夹。

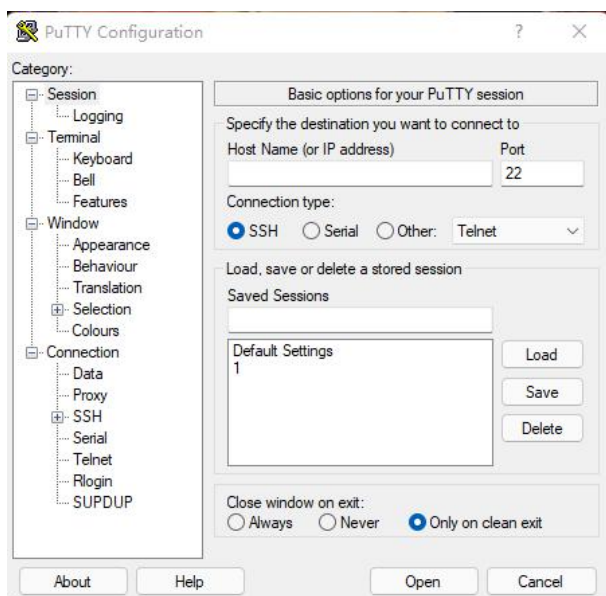
步骤 4 执行拷贝到目标储存路径。

4 通过串口工具 putty 抓取 log

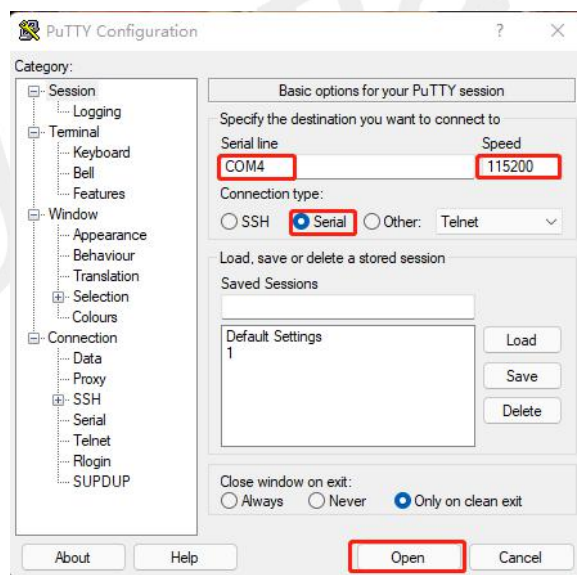
使用方法：

1 去 putty 官网下载对应位数的软件到本地

2 下载得到的直接是一个文件名为 putty.exe 的可执行文件，不需要安装即可使用。双击打开后，进入 Putty 的主界面。



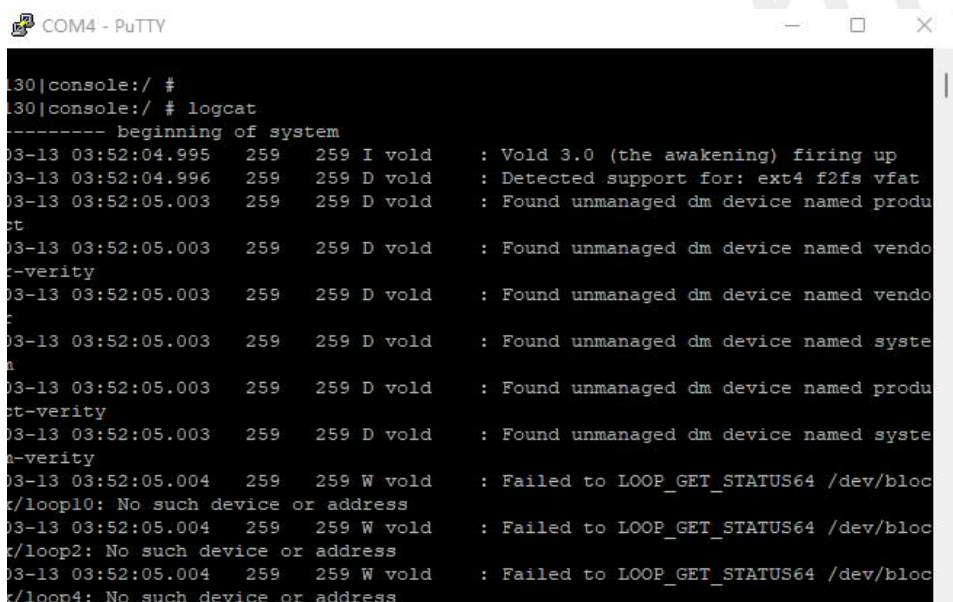
3 将设备用 Micro USB 线连接，添加可以识别的串口，将比特率修改为 115200，连接类型选择 Serial（串口）类型，然后点击 open



按 Enter 键验证是否连接



4 出现以上内容代表已经连接
输入 logcat 抓取 log。Ctrl+C 停止



5 不同场景下抓取 log 的方式

5.1 可以使用 adb（app 闪退、报错；系统开机问题：开机慢 动画相关问题）

直接使用 adb 工具抓取

5.2 无法使用 adb（死机；冻屏；黑屏；没有 usb 口）

5.2.1 使用串口工具

5.2.2 离线 log

adb shell

nohup logcat > data/anr/1.txt &

在 bug 复现之前输入，复现后导出即可

5.3 不需要使用 adb 工具

Ylog 独有

5.4 特殊情况（冻屏）

由于定屏类问题比较复杂，更多的是需要一些现场信息，为此需要进行以下操作来确认现场信息：

步骤 1 记录发生定屏的时间点和操作步骤，并进行拍照。

步骤 2 确认 adb 是否可以连接，在 cmd 命令行输入“adb device”，看是否可以打印出手机信息。

步骤 3 有 SIM 卡的情况下给测试机打电话，是否听到铃声，可以判断 modem 工作是否正常。

步骤 4 在 5 分钟之内，分别按 2~3 次“power key”以及音量键等物理按键，看看屏幕是否有变化。

步骤 5 若做完以上操作后，系统仍处于定屏状态，则需要抓取 sysdump log，通过 adb shell dumpsys activity 命令导出 Log

5.5 网络问题

tcpdump 是 linux 下的抓包工具，在 android 中没有，需要下载对应的工具。

下载地址：<https://www.androidtcpdump.com/android-tcpdump/downloads>

然后通过 adb 放到对应的目录：

adb root

adb push tcpdump /data/local/

可能会遇到无法 push 的情况，其实是没有权限，可以授权后重试：

adb shell

su

修改权限：

chmod 777 /data/local/tcpdump

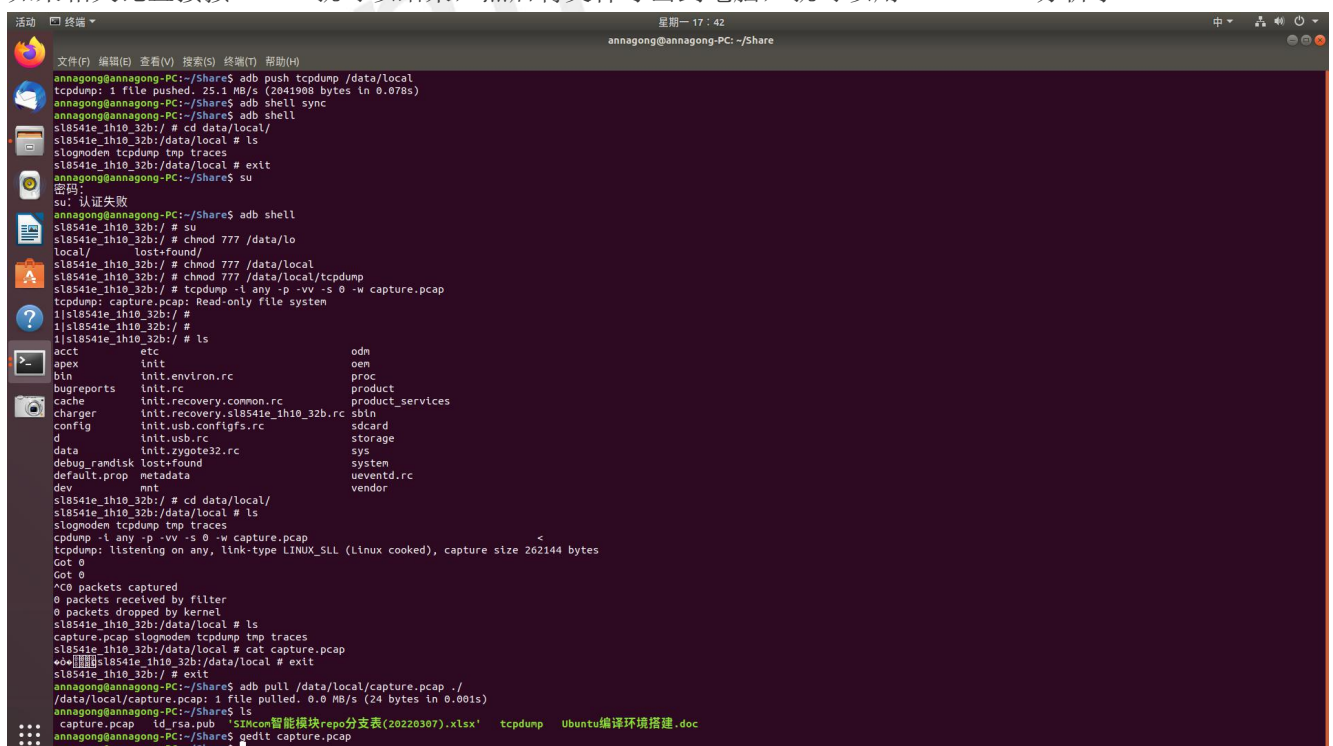
执行抓包命令：

cd data/local/

tcpdump -i any -p -vv -s 0 -w capture.pcap

启动抓包后会在当前目录创建一个 capture.pcap 的文件，会随着网络访问的次数增加来不断地增加文件大小。

如果相关比直接按 ctrl+c 就可以结束，然后将文件导出到电脑，就可以用 wireshark 分析了



```

annagong@annagong-PC: ~/Share
tcpdump: 1 file pushed, 25.1 MB/s (2041908 bytes in 0.078s)
annagong@annagong-PC: ~/Share$ adb shell sync
annagong@annagong-PC: ~/Share$ adb shell
s18541e_1h10_32b:/ # cd data/local/
s18541e_1h10_32b:/data/local # ls
slognode tcpdump tmp traces
s18541e_1h10_32b:/data/local # exit
annagong@annagong-PC: ~/Share$ su
密码:
su: 认证失败
annagong@annagong-PC: ~/Share$ adb shell
s18541e_1h10_32b:/ # su
s18541e_1h10_32b:/ # chmod 777 /data/lo
local/
s18541e_1h10_32b:/ # chmod 777 /data/local
s18541e_1h10_32b:/ # chmod 777 /data/local/tcpdump
s18541e_1h10_32b:/ # tcpdump -i any -p -vv -s 0 -w capture.pcap
tcpdump: capture.pcap: Read-only file system
1|s18541e_1h10_32b:/ #
1|s18541e_1h10_32b:/ #
1|s18541e_1h10_32b:/ # ls
acct          etc            odm
apex          init           oem
bin           init.environ.rc proc
bugreports   init.rc        product
cache        init.recovery.common.rc product_services
charger      init.recovery.s18541e_1h10_32b.rc sbin
config       init.usb.configfs.rc sdcard
d            init.usb.rc    storage
data         init.zygote32.rc sys
debug_ramdisk lost+found     system
default.prop metadata       ueventd.rc
dev          mnt            vendor
s18541e_1h10_32b:/ # cd data/local/
s18541e_1h10_32b:/data/local # ls
slognode tcpdump tmp traces
cpdump -i any -p -vv -s 0 -w capture.pcap
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
Got 0
Got 0
^C0 packets captured
0 packets received by filter
0 packets dropped by kernel
s18541e_1h10_32b:/data/local # ls
capture.pcap slognode tcpdump tmp traces
s18541e_1h10_32b:/data/local # cat capture.pcap
00000000 s18541e_1h10_32b:/data/local # exit
annagong@annagong-PC: ~/Share$ adb pull /data/local/capture.pcap ./
/data/local/capture.pcap: 1 file pulled, 0.0 MB/s (24 bytes in 0.001s)
annagong@annagong-PC: ~/Share$ ls
capture.pcap id_rsa.pub 'SIMCom智能模块repo分支表(20220307).xlsx' tcpdump Ubuntu编译环境搭建.doc
annagong@annagong-PC: ~/Share$ gedit capture.pcap
annagong@annagong-PC: ~/Share$

```

SIMCom
Confidential