**新国都Linux系统中串口升级问题分析**

文档版本：V1.0.0

更新日期：2016.12.07

版权声明

版权所有©2015 深圳市广和通无线股份有限公司。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

商标申明

为深圳市广和通无线股份有限公司的注册商标，由所有人拥有。



版本记录

|  |  |  |
| --- | --- | --- |
| 文档版本 | 更新日期 | 说明 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

适用型号

|  |  |  |
| --- | --- | --- |
| 序号 | 产品型号 | 说明 |
|  |  |  |
|  |  |  |

**目录**

[1.问题现象 4](#_Toc427591969)

[模块版本： 4](#_Toc427591970)

[使用场景： 4](#_Toc427591971)

[问题描述： 4](#_Toc427591972)

[2. 原因分析 5](#_Toc427591973)

[3. 解决方案及注意事项 6](#_Toc427591974)

[修改文件 6](#_Toc427591975)

[修改函数 6](#_Toc427591976)

[原始代码 6](#_Toc427591977)

[改后代码 6](#_Toc427591978)

# 1.问题现象

模块版本： H350 H330

使用场景： Linux 2.6.31 arm架构 飞思卡尔平台

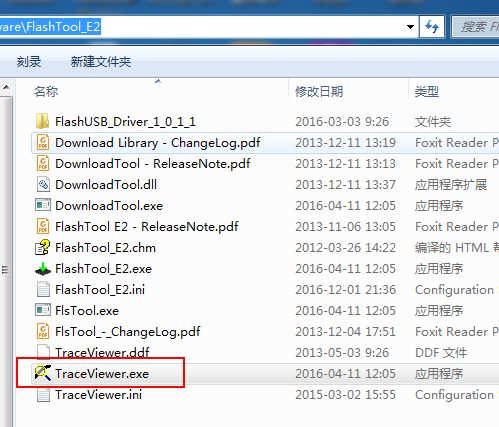
问题描述：

1. 通过物理串口，使用DownloadTool工具，进行升级

2. 在客户样机升级失败，在友善之臂开发板升级成功，在Ubuntu系统升级成功。

分析工具

1. TraceViewer.exe ，存放于windows Flash Tool工具目录



用于分析升级过程中的数据收发。

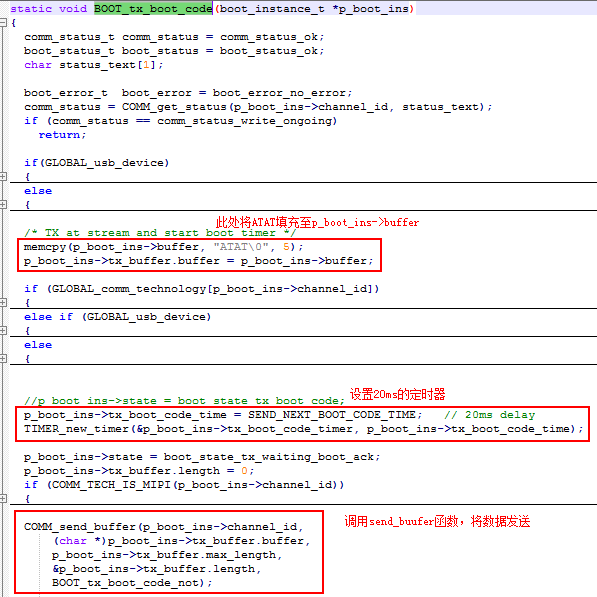
1. CommMonitor 串口监视精灵 6.1网上下载的一个串口监视软件，可以监控到串口上所有数据的收发

# 2. 问题分析

**客户POS机本例后续简称 异常机， Ubuntu系统本例后续简称 正常机**

1. . 使用串口精灵，监控物理串口的数据收发。对DownloadTool添加LOG打印，对比正常跟异常LOG数据。从串口精灵监控的数据收发看，在升级工具成功运行后，会提示“Please Reboot Your Phone Device”, 此时，正常机串口精灵会监控到一直在41 54 41 54数据循环发送，间隔大概率20ms一次，异常机只发送一次 41 54 41 54后，就停止发送了。41 54解码后为”ATAT“字串。
2. . 在DownloadTool工具源码，搜索ATAT关键字，查看代码流程。

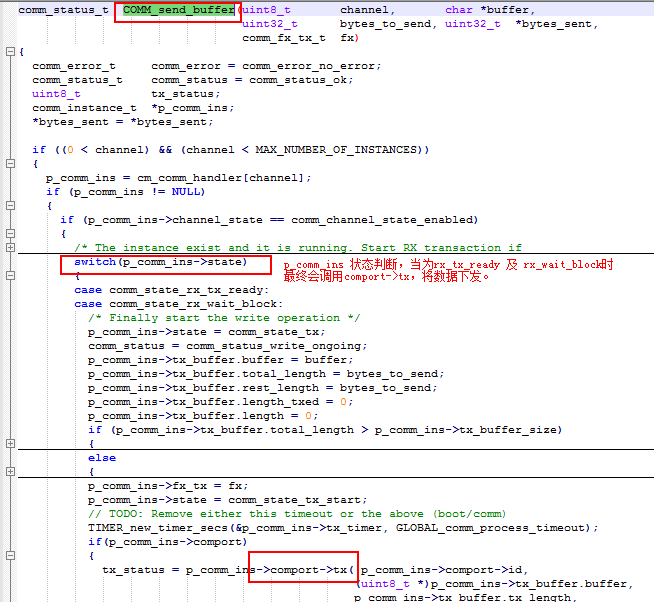
在msw\_tools\FlashTool\libDownloadTool\src\Portable\_code\boot\_handler\IFWD\_boot\_sgold.c文件的BOOT\_tx\_boot\_code函数查找到关键字。



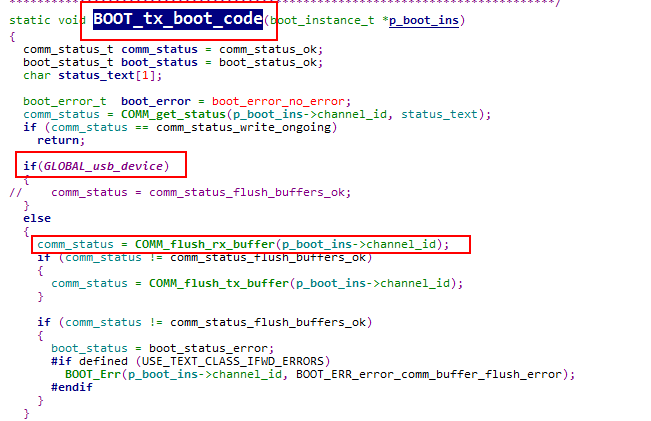
从此段代码看，只要流程正常，且定时器没关掉，ATAT将会不断发送，所以先查找定时器关掉的函数代码，其次要判断COMM\_send\_buffer是否最终有把数据发送。

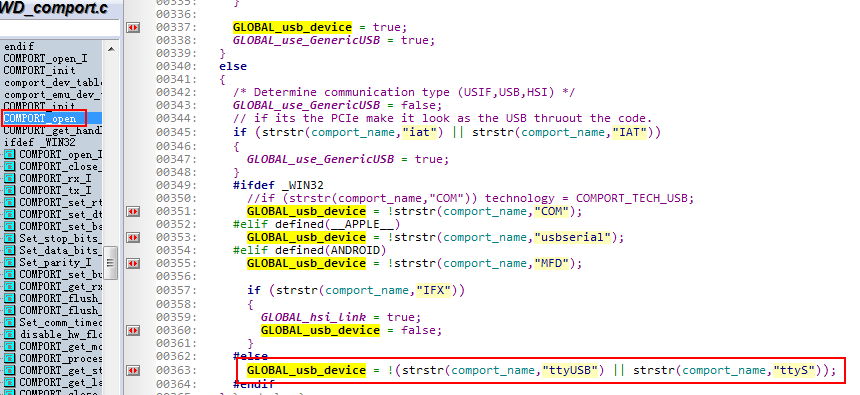
2.1). 根据TIMER\_new\_timer的命名规则，查找把关闭定时器函数名为TIMER\_stop\_timer,再找到tx\_boot\_code\_timer相关代码，并添加打印LOG，同时在BOOT\_tx\_boot\_code函数的new timer处添加打印，运行后发现BOOT\_tx\_boot\_code在不断运行，ATAT应会不断的填充并调用COMM\_send\_buffer函数，并进行下发。

2.2). 根据2.1)对COMM\_send\_buffer进行分析，



根据代码流程，对p\_comm\_ins->state的状态进行打印，并对comm\_state\_rx\_tx\_ready赋值处进行打印添加，编译运行后发现p\_comm\_ins->state第一次状态为comm\_state\_rx\_tx\_ready，但后继输出时状态为comm\_state\_rx\_in\_progress。另外对比正常机LOG发现，在调用COMM\_send\_buffer之前，状态都会被置成comm\_state\_rx\_tx\_ready，但异常机的则无此LOG输出。最终根据LOG打印，发现正常机在调用COMM\_send\_buffer之前会先调用COMM\_flush\_rx\_buffer，将状态转成comm\_state\_rx\_tx\_ready。详细查看COMM\_flush\_rx\_buffer被调用之处发现BOOT\_tx\_boot\_code会调用COMM\_flush\_rx\_buffer.



从上述代码流程看， COMM\_flush\_rx\_buffer的调用是由GLOBAL\_usb\_device全局变量决定，但此时使用的是串口，原则上此变量应该为FALSE，但从LOG输出来看，此变量为TRUE。搜索GLOBAL\_usb\_device赋值处代码;在COMPORT\_open处理，可看到红框处代码

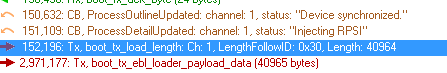
异常机当前使用的串口名称为ttymxc1, 正常机使用的串口名称为ttyUSB0, 红框处的判断，所以异常机GLOBAL\_usb\_device将被赋值为TRUE， 正常机将为FALSE。问题点已经找到，需要将异常机串口名称添加致红框代码处。修改后为：



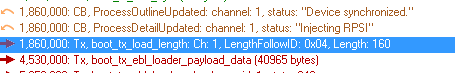
将此处修改后ATAT数据能持续发送。但后续升级依然失败。继续分析下一步数据。

1. 根据2的最后结论，升级依然失败，后续将会是升级的一系列交互数据。对于数据的分析将使用到TraceViewer工具。另DownloadTool在运行时可加入 -l 参数(DownloadTool -c /dev/ttyUSB0 -l trace.log H330\_V2H.00.09.fls)，抓取出 xxx.log文件，使用TraceViewer工具进行升级数据分析。抓取正常机与异常机的traceviewer文件。比对数据。

正常机在输出Device synchronized后，下发的第一组数据为 0x30, Length: 40964

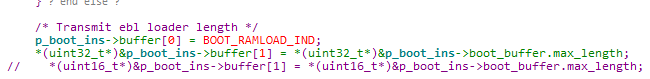


但异常机输出Device synchronized后，下发的第一组数据为0x04, Length:160.

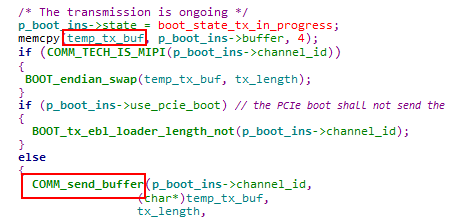


根据此两组数据对比及在正常机进行两次不同软件版本间的升级时,第一组数据值不会变(即保持为： LengthFollowdID: 0x30, Length: 40964. )的对比情况，怀疑异常机下发的第一组数据值存在丢失。

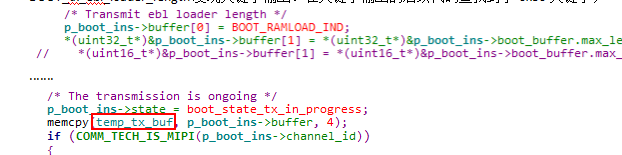
搜索Injecting RPSI关键字，在msw\_tools\FlashTool\libDownloadTool\src\Portable\_code\boot\_handler\IFWD\_boot\_sgold.c文件的BOOT\_tx\_ebl\_loader\_length发现关键字输出。在关键字输出的后续代码查找到了0x30关键字，



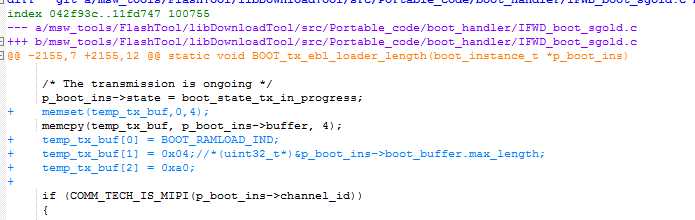
.......



从上述代码不难看出，程序先对p\_boot\_ins->buffer[0] [1]进行填充，其中buffer[0]为BOOT\_RAMLOAD\_IND(0x30)，buffer[1]为max\_length.后面会将p\_boot\_ins拷贝至temp\_tx\_buf，之后再调用COMM\_send\_buffer将temp\_tx\_buf发送。分别在memcpy前后对temp\_tx\_buf、p\_boot\_ins->buffer的每一组值进行打印，最终发现异常机memcpy前p\_boot\_ins->buffer的内容已经为0x04 0xA0,正常机memcpy前后数据保持为0x30, 0x04, 0xA0。再进一步添加打印发现异常机在第一次赋值到memcpy中间0x30数据被丢弃掉了。



因0x30在p\_boot\_ins->buffer中被改写或者丢弃，而最终发送的又是temp\_tx\_buf的数据，所以尝试将temp\_tx\_buf的数据写死。修改如下：



编译后升级程序成功运行，并且软件成功升级。

# 3. 解决方案及注意事项

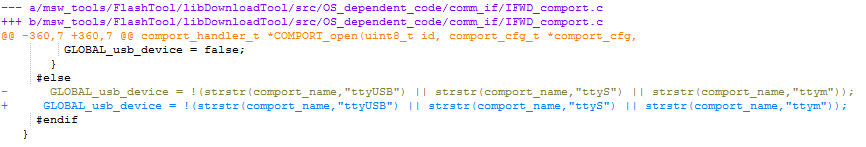
1. 修改

msw\_tools\FlashTool\libDownloadTool\src\OS\_dependent\_code\comm\_if\IFWD\_comport.c

msw\_tools\FlashTool\libDownloadTool\src\Portable\_code\boot\_handler\IFWD\_boot\_sgold.c

文件

1. 修改代码



.....

