# **#38172 4g/5g功能支持模组重启恢复需求文档**

# **1. 版本修订和Review记录**

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 版本编号 | 作者 | 说明 |
| 2022-12-01 | V1.0 | @赵昱钦 | 初稿 |
| 2022-12-05 | V1.1 | @赵昱钦 | 按照建议修改、增加crash日志收集 |
| 2022-12-26 | v1.2 | @赵昱钦 | 功能点3，Orch告警部分 |

# **2. 术语**

ECM CPE通过唤起该进程进行拨号，进而网络通信

FM150 LW2508、LW2509平台5G无线模块

FM150模块AT：

AT+DUMPEN=1 收集dump文件前需配置该AT，一般用在crash日志分析时

at$qcrmcall=1,1,1 拨号AT cmd

# **3. 需求场景描述**

### **一：背景**

当4g/5g模块（FM150、EC20等）发生crash，4g/5g接口会消失并重新创建，导致网络中断，需要用户重启设备才能恢复。

我们很多线上bug，都来源于模块crash，crash的原因需抓dump日志反馈厂商分析，crash发生后用户需要手动重启，网络才能恢复正常，对于仅使用4G/5G唯一网口的客户影响就较大。

### **二：期望**

1、能够对模块crash引起的接口消失重建做出识别，并实现自动恢复网络的效果。

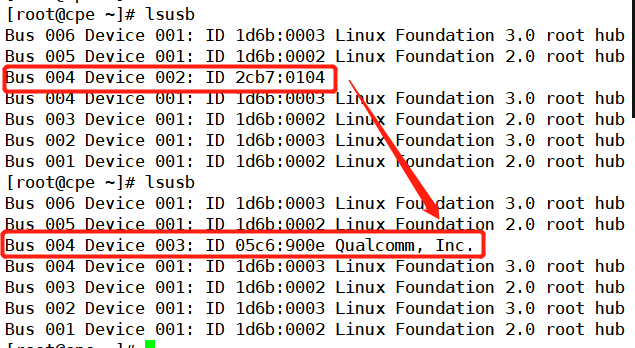
2、增加crash日志自动收集

# **4. 功能设计描述**

### **一、如何模拟crash场景**

#### FM150模块:

通过adb工具（文档"其他"中有该工具），输入adb shell，进入模块后再输入echo c > /proc/sysrq-trigger，lsusb查看2cb7:0104变为05c6:900e时，说明模块出现了crash，如下图：



注：仅模拟模块crash，不需要输入AT+DUMPEN=1

细节参考：<https://project.appexnetworks.com.cn/boards/32/topics/1789>

注：at+gtusbmode=17时支持adb工具，32不支持。

#### 其他模块：

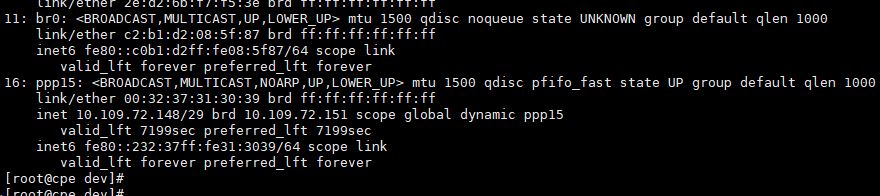
待补充

### **二、怎么支持模块重启恢复：**

一般4G/5G拨号是经历**AT拨号，dhcp**，dhclient进程将ip等信息上报三个过程；重启恢复主要关注前两阶段。

**1、crash后现象**

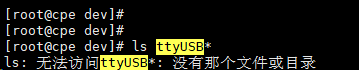
A）接口消失，一段时间后恢复，但为DOWN





B）对应的串口文件ttyUSBn消失，一段时间后恢复



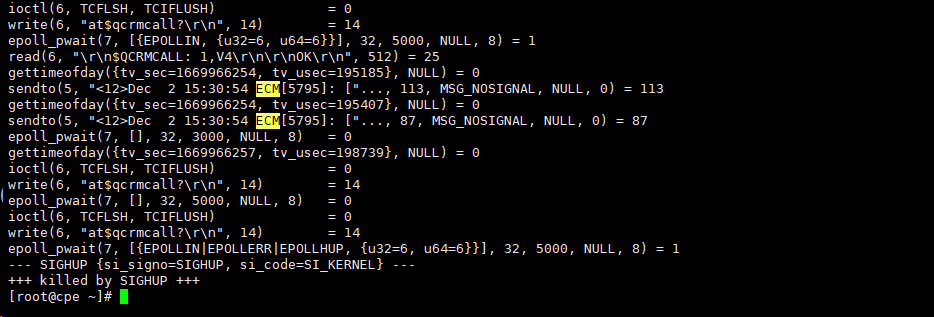


C）ECM拨号进程挂掉。

**2、crash后接口down问题**

驱动重新加载后，先up接口，再进行AT命令与dhcp。

**3、crash后ECM进程挂掉问题**

通过strace -p $pid测试ECM进程在模块crash时，会收到一个SIGHUP信号，从而退出，如下：

**4、ECM进程如何判断是模块crash？**

使用libudev库相关api，设备状态发生变化时，事件通知。

**5、crash日志收集**

#### FM150模块:

收集命令：

echo -e "05c6 900e" > /sys/bus/usb-serial/drivers/option1/new\_id  
 ./fibo\_dump\_collect\_tool -f ./dump\_log(dump\_log不需要手动创建)

细节参考：<https://project.appexnetworks.com.cn/boards/32/topics/1789> **LW2508 5G模块收集dump文件部分**

#### **其他模块：**

待补充

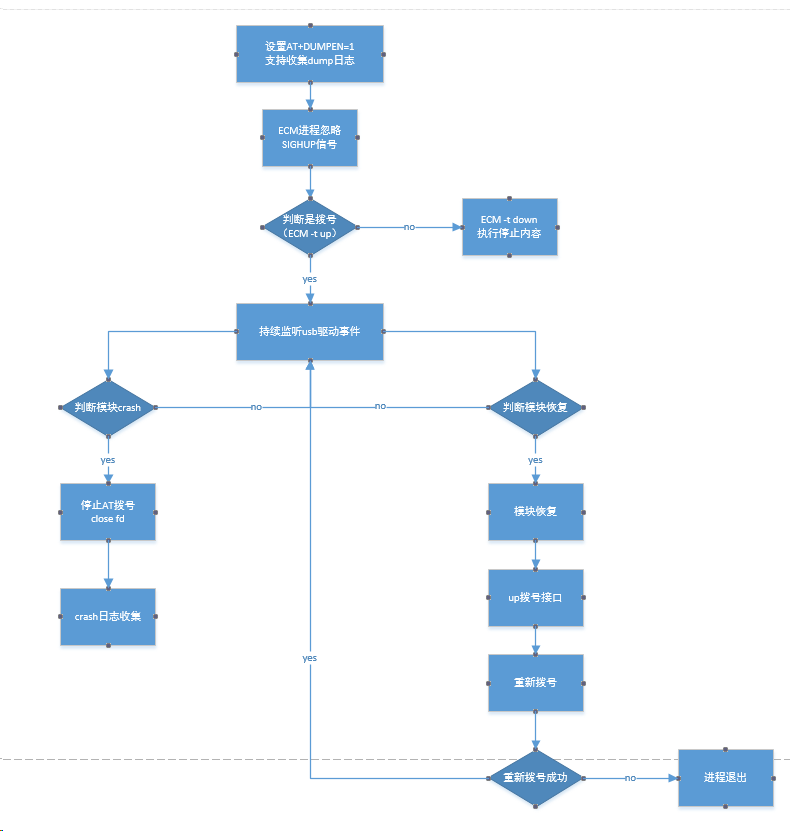
6、crash后，恢复拨号及过程

ECM进程忽略SIGHUP信号，保证异常情况不退出，判断crash现象，认为是一次crash后，进行crash日志收集，待接口及串口文件重新加载后，再执行AT和dhcp。

## **4.1 功能整体逻辑简述和架构设计图**

恢复拨号流程图：

#### FM150模块:



注，判断模块crash和模块恢复，参考**4.2.1**

#### 其他模块：

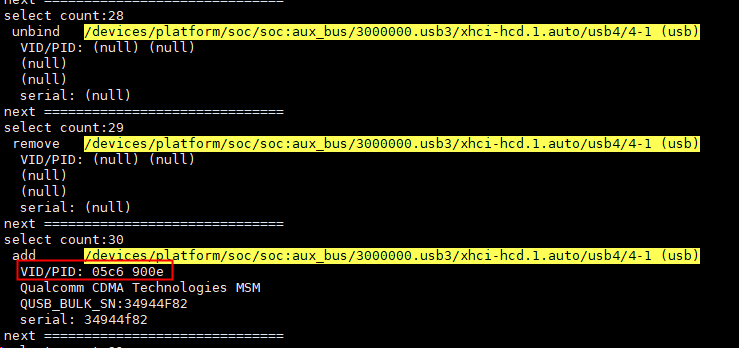
待补充

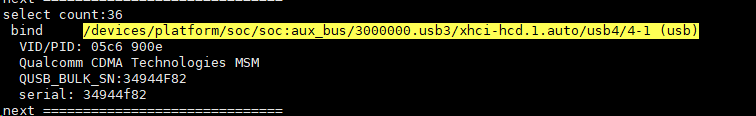
## **4.2 功能细节逻辑描述**

### **4.2.1 功能点1，**监听模块crash/恢复

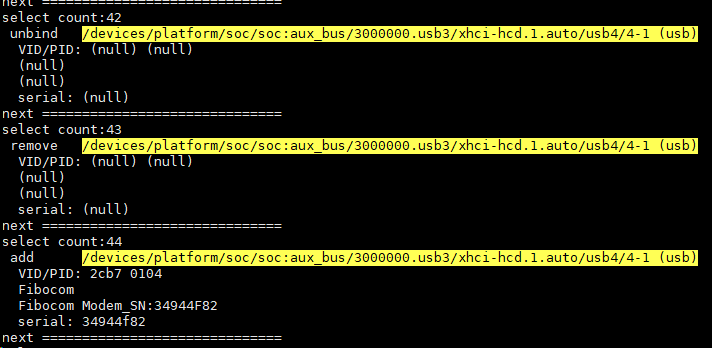
通过demo测试，可检测到add、remove、unbind、bind等事件，从而让进程关注到设备情况。

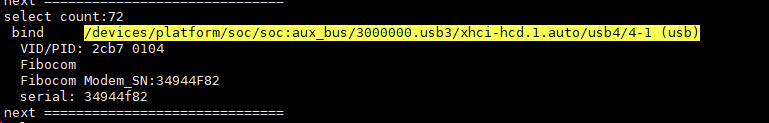
crash时，会unbind、remove，再add和bind VID/PID：05c6 900e【需设AT+DUMPEN=1，否则不会add和bind VID/PID：05c6 900e】。





模块自动恢复时，会unbind、remove，再add和bind VID/PID：2cb7 0104





ECM进程监听bind VID/PID：05c6 900e事件，认为这是一次crash发生；监听bind VID/PID：2cb7 0104事件，认为这是模块恢复，能够重新拨号恢复网络。

### 4.2.2 功能点2，收集dump日志

#### FM150模块:

实现逻辑：ECM进程使用/bin/appex/fibo\_dump\_collect\_tool -f /var/log/dump\_log 收集dump日志，日志存储在/var/log/dump\_log文件夹，多次收集日志时覆盖上一次收集。

注：fibo\_dump\_collect\_tool 工具把模组dump压缩后导出，不超过150MB。

#### 其他模块：

待补充

### **4.2.3 功能点3，Orch告警**

ECM\_DEMO拨号进程在检测到Crash后，会整合信息通过json发往agent进程，agent再通过protobuf消息发往Orch告警。

1、ECM\_DEMO进程通过/bin/appex/cpeagent cmdline方式组织json发往agent

示例：

{  
"WwanManagerInfo": {  
"InfoType": "1",  
"WwanModule": "3",  
"CrashInfo": {  
"TimeStamp": "1672211355",  
"CrashFileFlag": "1",  
"CrashFileName": "/var/log/dump\_log"  
}  
}  
}

WwanManagerInfo：消息结构体

InfoType：整数类型，消息类型，1为crash消息

WwanModule：枚举型，无线模块类型，在lightwanMsg.h中LW\_WWAN\_MODULE\_MODEL\_TYPE定义

CrashInfo：crash消息结构体

TimeStamp：整数类型，模块crash时间戳，单位秒

CrashFileFlag: 整数类型，1 crash日志收集成功，0 失败

CrashFileName：字符串，crash日志目录

2、agent解析json，组织消息发往Orch，protobuf消息如下：

message InterfaceReport\_C2O {  
 required uint32 ifStatus = 1; // 表示LW\_IF\_STATUS中LW\_IF\_STATUS\_WWAN\_CHANGED=5  
 required uint32 wanId = 2;// 表示wan id  
 required string ifName = 3;// 表示ppp15接口  
 repeated Cidr ipList = 4;//NULL  
 optional uint32 seqID = 5;// 表示sequence id  
 repeated uint32 dnsList = 6;//NULL  
 optional uint32 gateway = 7;//NULL  
 optional uint32 lanId = 8;//NULL  
 repeated Ipv6Cidr ipv6List = 9;//NULL  
 repeated bytes ipv6DnsList = 10;//NULL  
 optional bytes ipv6Gateway = 11;//NULL  
 optional WWanAlert wwanAlert = 12;// 结构体  
}  
  
message WWanAlert {  
 required uint32 alertType = 1; // 表示告警类型，LW\_WWAN\_ALERT\_TYPE，目前仅有crash类型  
 required uint32 moduleModel = 2; // 表示模块型号，LW\_WWAN\_MODULE\_MODEL\_TYPE  
 optional WWanAlertCrash wwanAlertCrash = 3;// crash类型结构体  
}  
  
message WWanAlertCrash {  
 required uint32 timeStamp = 1; // 表示crash发生时间戳，seconds  
 required uint32 crashFileFlag = 2; // 表示crash日志收集错误码，0 不需要收集日志，1成功，2失败  
 optional string crashFileName = 3; // 表示crash日志收集目录，需要crashFileFlag等于1。  
}

3、告警需确保Orch收到

对于单独4G/5G作唯一WAN口场景，crash期间网络会异常，CPE需要Orch的reply来确定Orch收到了消息；使用413/414发送接收消息，CPE收到414消息停止上报，参考2中的protobuf定义。

注：模块每次发生crash，都将组成一条消息，间隔20秒上报10次；如果10次都没收到Orch回复，日志打印"already reported 10 times"停止上报。

示例：

场景一： 仅5G作为WAN口，模块发生crash。

[2023/02/06\_19:07:15.149][INFO][\_LW\_JsonToWwanManagerInfo-257]Wwan Manager crash information, InfoType:1 WwanModule:3 TimeStamp:1675681575 CrashFileFlag:1 CrashFileName:/var/log/dump\_log  
[2023/02/06\_19:07:35.152][ERROR][\_LW\_WwanManagerReportSingle-426]Wwan Manager report failed, (-22:Invalid argument)  
[2023/02/06\_19:07:55.151][INFO][\_LW\_FormatPrintMsg-372]Send Msg: Type = 413 netId:0 transactionId:168 msgBase{ interfaceReport{ ifStatus:5 wanId:2 ifName:"ppp15" seqID:1675681635 wwanAlert{ alertType:1 moduleModel:3 wwanAlertCrash{ timeStamp:1675681575 crashFileFlag:1 crashFileName:"/var/log/dump\_log" } } } }   
[2023/02/06\_19:07:56.345][INFO][\_LW\_FormatPrintMsg-372]Recv Msg: Type = 414 netId:0 transactionId:168 errorCode:0 msgBase{ interfaceReport{ ifStatus:5 wanId:2 ifName:"ppp15" seqID:1675681635 wwanAlert{ alertType:1 moduleModel:3 wwanAlertCrash{ timeStamp:1675681575 crashFileFlag:1 crashFileName:"/var/log/dump\_log" } } } }

场景二： 告警10次上报都没收到Orch回复。

[2023/02/08\_14:18:29.693][INFO][\_LW\_FormatPrintMsg-372]Send Msg: Type = 413 netId:0 transactionId:34 msgBase{ interfaceReport{ ifStatus:5 wanId:2 ifName:"ppp15" seqID:1675837109 wwanAlert{ alertType:1 moduleModel:3 wwanAlertCrash{ timeStamp:1675655897 crashFileFlag:1 crashFileName:"/var/log/dump\_log1" } } } }   
[2023/02/08\_14:18:34.696][INFO][\_LW\_FormatPrintMsg-372]Send Msg: Type = 413 netId:0 transactionId:38 msgBase{ interfaceReport{ ifStatus:5 wanId:2 ifName:"ppp15" seqID:1675837109 wwanAlert{ alertType:1 moduleModel:3 wwanAlertCrash{ timeStamp:1675655897 crashFileFlag:1 crashFileName:"/var/log/dump\_log1" } } } }   
[2023/02/08\_14:18:39.698][INFO][\_LW\_FormatPrintMsg-372]Send Msg: Type = 413 netId:0 transactionId:43 msgBase{ interfaceReport{ ifStatus:5 wanId:2 ifName:"ppp15" seqID:1675837109 wwanAlert{ alertType:1 moduleModel:3 wwanAlertCrash{ timeStamp:1675655897 crashFileFlag:1 crashFileName:"/var/log/dump\_log1" } } } }   
[2023/02/08\_14:18:44.702][INFO][\_LW\_FormatPrintMsg-372]Send Msg: Type = 413 netId:0 transactionId:47 msgBase{ interfaceReport{ ifStatus:5 wanId:2 ifName:"ppp15" seqID:1675837109 wwanAlert{ alertType:1 moduleModel:3 wwanAlertCrash{ timeStamp:1675655897 crashFileFlag:1 crashFileName:"/var/log/dump\_log1" } } } }   
[2023/02/08\_14:18:49.705][INFO][\_LW\_FormatPrintMsg-372]Send Msg: Type = 413 netId:0 transactionId:51 msgBase{ interfaceReport{ ifStatus:5 wanId:2 ifName:"ppp15" seqID:1675837109 wwanAlert{ alertType:1 moduleModel:3 wwanAlertCrash{ timeStamp:1675655897 crashFileFlag:1 crashFileName:"/var/log/dump\_log1" } } } }   
[2023/02/08\_14:18:54.707][INFO][\_LW\_FormatPrintMsg-372]Send Msg: Type = 413 netId:0 transactionId:55 msgBase{ interfaceReport{ ifStatus:5 wanId:2 ifName:"ppp15" seqID:1675837109 wwanAlert{ alertType:1 moduleModel:3 wwanAlertCrash{ timeStamp:1675655897 crashFileFlag:1 crashFileName:"/var/log/dump\_log1" } } } }   
[2023/02/08\_14:18:59.710][INFO][\_LW\_FormatPrintMsg-372]Send Msg: Type = 413 netId:0 transactionId:59 msgBase{ interfaceReport{ ifStatus:5 wanId:2 ifName:"ppp15" seqID:1675837109 wwanAlert{ alertType:1 moduleModel:3 wwanAlertCrash{ timeStamp:1675655897 crashFileFlag:1 crashFileName:"/var/log/dump\_log1" } } } }   
[2023/02/08\_14:19:04.713][INFO][\_LW\_FormatPrintMsg-372]Send Msg: Type = 413 netId:0 transactionId:63 msgBase{ interfaceReport{ ifStatus:5 wanId:2 ifName:"ppp15" seqID:1675837109 wwanAlert{ alertType:1 moduleModel:3 wwanAlertCrash{ timeStamp:1675655897 crashFileFlag:1 crashFileName:"/var/log/dump\_log1" } } } }   
[2023/02/08\_14:19:09.716][INFO][\_LW\_FormatPrintMsg-372]Send Msg: Type = 413 netId:0 transactionId:68 msgBase{ interfaceReport{ ifStatus:5 wanId:2 ifName:"ppp15" seqID:1675837109 wwanAlert{ alertType:1 moduleModel:3 wwanAlertCrash{ timeStamp:1675655897 crashFileFlag:1 crashFileName:"/var/log/dump\_log1" } } } }   
[2023/02/08\_14:19:14.718][INFO][\_LW\_FormatPrintMsg-372]Send Msg: Type = 413 netId:0 transactionId:71 msgBase{ interfaceReport{ ifStatus:5 wanId:2 ifName:"ppp15" seqID:1675837109 wwanAlert{ alertType:1 moduleModel:3 wwanAlertCrash{ timeStamp:1675655897 crashFileFlag:1 crashFileName:"/var/log/dump\_log1" } } } }   
[2023/02/08\_14:19:19.719][WARN][\_LW\_WwanManagerReport-474]Wwan manager SeqId:1675837109 already reported 10 times

# **5. UI交互设计**

不涉及

# **6. 诊断调试功能**

## **6.1 log日志**

#### FM150模块:

收集dump日志/var/log/dump\_log

#### 其他模块：

待补充

# **7. 产品规格**

## **7.1 支持平台**

功能特性1：

|  |  |  |
| --- | --- | --- |
| **设备型号** | **支持**  **(是/否/不涉及)** | **备注** |
| LW1305 |  |  |
| LW2X02 |  |  |
| LW2X08 | 是 | 仅LW2508（centos arm64） |
| LW2X09 | 是 | 仅LW2509 (centos x86) |
| LW300X |  |  |
| LW700X |  |  |
| vCPE |  |  |
| PoP |  |  |
| Windows |  |  |
| Mac |  |  |
| ios |  |  |
| Android |  |  |
| SDK（嵌入式设备） |  |  |

## **7.2 规格列表**

**不涉及**

# 8. 功能缺陷或遗留问题

# **9. 测试建议**

#### FM150模块:

1、模块crash期间，监听模块热插拔、收集dump日志、告警、拨号恢复是否正常

2、测试LW2508、LW2509两个平台

#### 其他模块：

待补充

# **10. 其它**

[请至钉钉文档查看附件《NRCHECK\_TOOL\_V1.0.5.tar.gz》](https://alidocs.dingtalk.com/i/nodes/3xRN9bGQyw4JbxjgrlyzWzXPADKnorv6?doc_type=wiki_doc&iframeQuery=anchorId%253DX02lodxep566xkppfp6zjk)

NRCHECK\_TOOL内含adb工具（模拟FM150-AE/FM160及子模块模块crash）

# 11. 会议纪要

# **12.** 相关文档