

MEIG-SLM3XX系列RIL 调试方法和常见问题

发布日期：2020年7月9日

受控文件名称：MEIG-SLM3XX系列_RIL_调试方法和常见问题

受控版本号：V1.00

发布机构：美格智能技术股份有限公司数据事业部



重要声明

版权声明

版权所有：美格智能技术股份有限公司

本资料及其包含的所有内容为美格智能技术股份有限公司所有，受中国法律及适用之国际公约中有关著作权法律的保护。未经美格智能技术股份有限公司书面授权，任何人不得以任何形式复制、传播、散布、改动或以其它方式使用本资料的部分或全部内容，违者将被依法追究责任。

不保证声明

美格智能技术股份有限公司不在此文档中的任何内容作任何明示或暗示的陈述或保证，而且不对特定目的的适销性及适用性或者任何间接、特殊或连带的损失承担任何责任。

保密声明

本文档（包含任何附件）包含的信息是保密信息。接收人了解其获得的本文档是保密的，限用于规定的目的外不得用于任何目的，也不得将本文档泄露给任何第三方。

免责声明

本公司不承担由于客户不正常操作造成的财产或者人身伤害责任。请客户按照手册中的技术规格和参考设计开发相应的产品。在未声明之前，本公司有权根据技术发展的需要对本手册内容进行更改，且更改版本不另行通知。

修改记录	
V1.00	首次发布
V1.01	修改文档名称

目录

RIL 适配常见问题.....	错误！未定义书签。
目录.....	4
1. 概述.....	5
2. 验证 RIL 合入正确.....	5
2.1 验证设备中端口.....	5
2.2 验证是否调用 meigril.....	6
2.3 验证上网功能.....	7
3. Android 端问题.....	7
3.1 通用问题.....	7
3.2 Android4.x.....	9
3.2 android5.x.....	10
3.3 android6.x.....	10
3.4 android7.x.....	10
3.5 android8.x.....	10
3.6 android9.x.....	10
4. 抓取 RIL log.....	10

1. 概述

本文档主要针对美格SLM3XX模块基于Android系统的适配指导说明，主要面向集成美格SLM3XX模块到Android设备上的相关开发调试人员。引导其快速适配美格模块到Android设备上。满足美格模块提供数据，语音，短信等电信业务给Android设备的要求。

2. 验证RIL合入正确

2.1 验证设备中端口

RIL适配合入后需要验证android侧是否有端口出来，验证方式如下：
dev目录下有节点ttyUSB0-5出现。

```
ttyGS1
ttyGS2
ttyGS3
ttyS0
ttyS1
ttyS2
ttyS3
ttyUSB0
ttyUSB1
ttyUSB2
ttyUSB3
ttyUSB4
ttyUSB5
tun
uhid
uinput
urandom
usb_accessory
vcs
vcs1
vcsa
vcsa1
watchdog
watchdog0
watchdog1
xt_qtaguid
zero
root@tulip-t1:/dev #
root@tulip-t1:/dev #
root@tulip-t1:/dev #
root@tulip-t1:/dev #
root@tulip-t1:/dev #
```

2.2 验证是否调用meigril

方法1：查看属性

```
# getprop | grep ril.libpath
[ril.libpath]: [libmeig-ril.so]
```

如果不是libmeig-ril.so，检查device目录下是否有其它代码定义了ril.libpatch的定义，有的话则删除。

方法2：添加log打印确保调用到了meigril

修改文件hardware/ril/rild/rild.c的main函数加入

```
int n;
for(n = 0; n < argc; n++) {
    RLOGD("**RILd param argc = %d, argv = %s", n, argv[n]);
}
```

正确情况打印log根据android版本不同大致如下：

Android9之前

```
D/RILD    ( 119): **RILd param argc = 0, argv = /system/bin/rild
D/RILD    ( 119): **RILd param argc = 1, argv = -l
D/RILD    ( 119): **RILd param argc = 2, argv = /system/lib/libmeig-ril.so
```

Android9以上

```
RILD      : **RIL Daemon Started**
RILD      : **RILd param count=3**
RILD      : **RILd param argc = 0, argv = /vendor/bin/hw/rild
RILD      : **RILd param argc = 1, argv = -l
RILD      : **RILd param argc = 2, argv = /vendor/lib64/libmeig-ril.so
```

2.3 验证上网功能

美格SLM3XX设备和Android系统连接好后上电开机，同时确保美格SLM3XX已插入有效SIM卡。adb shell进入android 上位机，正常情况下，通过ifconfig命令我们将看到ppp0或Eth1的网卡上有相应的ip。在有线及wifi断开的情况下ping 8.8.8.8能通，Android 系统能上网。

3. Android端问题

本章节的问题是由于市面上各版本各家的android设备都有一些定制，本章主要是美格在适配SLM3XX调试部分机型时候遇到的问题记录。

3.1 通用问题

1. 设置适合的网络制式

定义 ro.telephony.default_network 为 20 。 或者 修改 frameworks/base/telephony/java/com/android/internal/telephony/RILConstants.java 修改 int PREFERRED_NETWORK_MODE =

```
SystemProperties.getInt("ro.telephony.default_network",
NETWORK_MODE_WCDMA_PREF);
SystemProperties.getInt("ro.telephony.default_network",
NETWORK_MODE_LTE_TDSCDMA_GSM_WCDMA
```

改

为

2. 由于SLM3XX不支持CDMA防止android端误注册导致不能识别SIM卡

frameworks/opt/telephony/src/java/com/android/internal/telephony/uicc/IccCardProxy.java文件updateIccAvailability函数中加入以下patch:

```
-- a/android/frameworks/opt/telephony/src/java/com/android/internal/telephony/uicc/IccCardProxy.java
+++ b/android/frameworks/opt/telephony/src/java/com/android/internal/telephony/uicc/IccCardProxy.java
@@ -354,6 +354,8 @@ public class IccCardProxy extends Handler implements IccCard {
    }
}

+    if(mCurrentAppType == UiccController.APP_FAM_3GPP2) return;
+
    if (mIccRecords != newRecords || mUiccApplication != newApp || mUiccCard != newCard) {
        if (DBG) log("Icc changed. Reregistering.");
        unregisterUiccCardEvents();
@@ -529,6 +531,8 @@ public class IccCardProxy extends Handler implements IccCard {
    }
}
```

3. 确保android设备打开了语音和短信feature, 修改以下两个属性为true。注意: framework中的可能被device下的overlay。

```
<!-- This device is not "voice capable"; it's data-only. -->
- <bool name="config_voice_capable">false</bool>
+ <bool name="config_voice_capable">true</bool>

<!-- This device does not allow sms service. -->
- <bool name="config_sms_capable">false</bool>
+ <bool name="config_sms_capable">true</bool>
```

4. 有时候出现了pppd 拨号失败, 需要确保pppd 和 chat是同一份代码编译出来的并一起使用。最终会在out目录下生成chat和pppd的可执行程序。可以使用find指令查找确定已经生成。

chat	编译生成chat	2015/12/4 23:17	文件夹
pppd	编译生成pppd	2015/12/4 23:17	文件夹
CleanSpec.mk		2015/12/4 23:17	MK 文件

3 KB 确保都参加编译并配套使用

5. 部分机型设置SIM PIN之后, 开机不显示SIM PIN。发现为拿到的代码被产商修改, 改出来的问题。

frameworks/opt/telephony/src/java/com/android/internal/telephony/SubscriptionController.java

```
private boolean isSubInfoReady() {
+    /**
    boolean config = SystemProperties.getBoolean("ro.radio.noril", false);
    if(config == true) {
        return sSlotIndexToSubId.size() > 0;
@@ -222,7 +223,8 @@ public class SubscriptionController extends ISub.Stub {
        return true;
    }
    return false;
+    /**
+    return sSlotIndexToSubId.size() > 0;
+    */
}
```

改成android原始代码, 拿到的代码做了修改增加了卡状态ready判断, 少了pin判断

3.2 Android4. x

1. 报错: trySetupData: X retValue=false

APN配置错误, 需要导入平台正确的apns-conf.xml。

2. 对于有些SIM卡返回IP地址的时候形态如下需修改
frameworks/opt/telephony/src/java/com/android/internal/telephony/dataconnection/DataCallResponse.java

addresses=[IPv4:10.16.100.80,,, IPv6:::1:0:AFF2:13C6]修改

```

.....try{
.....//<!-- add by meig for Non-numeric ip addr start
.....//sample.IPV4:10.1.209.87-->.10.1.209.87
.....if(addr.startsWith("IPv4:") || addr.startsWith("IPv6:")){
.....    addr = addr.substring("IPv4:".length());
.....}
.....//add by meig for Non-numeric ip addr end-->
.....ia = NetworkUtils.numericToInetAddress(addr);
.....} catch (IllegalArgumentException e) {
.....    throw new UnknownHostException("Non-numeric ip addr=" + addr);
.....}
.....if (!ia.isAnyLocalAddress()) {
.....    if (addr.startsWith("IPv4:")) {

```

3. 短信开机空指针崩溃log: java.lang.RuntimeException: Unable to start receiver com.android.mms.transaction.MmsSystemEventReceiver:
java.lang.NullPointerException

修改: 确保networkAttributes数组中有短信能力, 我们这边的板子framework中有配置, 在device中的没有配置覆盖了framework。

```

@@ -81,6 +81,7 @@
<string-array translatable="false" name="networkAttributes">
    <item>"wifi,1,1,1,-1,true"</item>
    <item>"mobile,0,0,0,-1,true"</item>
+   <item>"mobile_mms,2,0,2,60000,true"</item>
    <item>"mobile_bluetooth,7,7,1,-1,true"</item>
    <item>"bluetooth,7,7,1,60000,true"</item>
    <item>"ethernet,9,9,2,-1,true"</item>

```

4. LTE信号问题: android4.4对于LTE信号的判断比较老。之后LTE信号信号全部都用RSRP来

判断, 所以要删除对于LTESignalStrength的check。修改文件
frameworks/base/telephony/java/android/telephony/SignalStrength.java

```
@@ -744,10 +744,12 @@ public class SignalStrength implements Parcelable {
    * dB= Number of Resource blocksxRSRP/RSSI SNR = gain=signal/noise ratio
    * = -10log P1/P2 dB
    */
+
+    /** remove by meig for android4.4 LTE signalstrength
+    if(mLteSignalStrength == 99){
+        if(DBG)log(" add mLteSignalStrength check");
+        return SIGNAL_STRENGTH_NONE_OR_UNKNOWN;
+    }
+    */
    int rssiIconLevel = SIGNAL_STRENGTH_NONE_OR_UNKNOWN, rsrpIconLevel = -1, snrIconLevel = -1;
    if (mLteRsrp > -44) rsrpIconLevel = -1;
@@ -781,7 +783,7 @@ public class SignalStrength implements Parcelable {
    * associated with LTE RSRP and the bars associated with the LTE
    * RS_SNR
    */
    return (rsrpIconLevel < snrIconLevel ? rsrpIconLevel : snrIconLevel);
+    return (rsrpIconLevel > snrIconLevel ? rsrpIconLevel : snrIconLevel); //modfiy by meig lte use rsrp
}
if (snrIconLevel != -1) return snrIconLevel;
```

3.2 android5.x

1. 不显示4G图标，showDataIcon判断条件异常。使用了错误的inet来判断是否显示dataicon

```
index dd31798..ea34083 100755
--- a/frameworks/base/packages/SystemUI/src/com/android/systemui/statusbar/policy/NetworkControllerImpl.java
+++ b/frameworks/base/packages/SystemUI/src/com/android/systemui/statusbar/policy/NetworkControllerImpl.java
@@ -1229,8 +1229,13 @@ public class NetworkControllerImpl extends BroadcastReceiver
    String contentDescription = getStringIfExists(getContentDescription());
    String dataContentDescription = getStringIfExists(icons.mDataContentDescription);

    boolean showDataIcon = mCurrentState.dataConnected && mCurrentState.inetForNetwork != 0
+    boolean showDataIcon = mCurrentState.dataConnected /**&& mCurrentState.inetForNetwork != 0 */
    || mCurrentState.iconGroup == TelephonyIcons.ROAMING;

    .....//.For mobile data, use general inet condition for phone signal indexing,
    .....//.and network specific for data indexing. (I think this might be a bug, but
    .....//.keeping for now).
    .....//.TODO: Update with explanation of why.
    .....mCurrentState.inetForNetwork = inetConditionForNetwork;
    .....setInetCondition(inetCondition);
```

3.3 android6.x

调试过程中除通用修改，没有遇到特殊需要修改的。

3.4 android7.x

调试过程中除通用修改，没有遇到特殊需要修改的。

3.5 android8.x

调试过程中除通用修改，没有遇到特殊需要修改的。

3.6 android9.x

调试过程中除通用修改，没有遇到特殊需要修改的。

6. 抓取RIL log

在windows cmd命令下执行logcat，如下图所示：

```
#抓取RIL模块的log
adb logcat -b radio -v time

#抓取整个Android系统的log
adb logcat -v time

#在usb端口的情况，logcat运行在设备后台抓取log
adb shell
logcat -b radio -v time > <filename> &
#测试结束后再把保存在设备上的log 移到本地电脑
adb pull <filename> <local directory>
```