

MEIG-SLM3XX系列RIL Android7.x适配文档

发布日期：2020年7月9日

受控文件名称：MEIG-SLM3XX系列_RIL_Android7.x适配文档

受控版本号：V1.01

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对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	修改文件名称

目录

目录

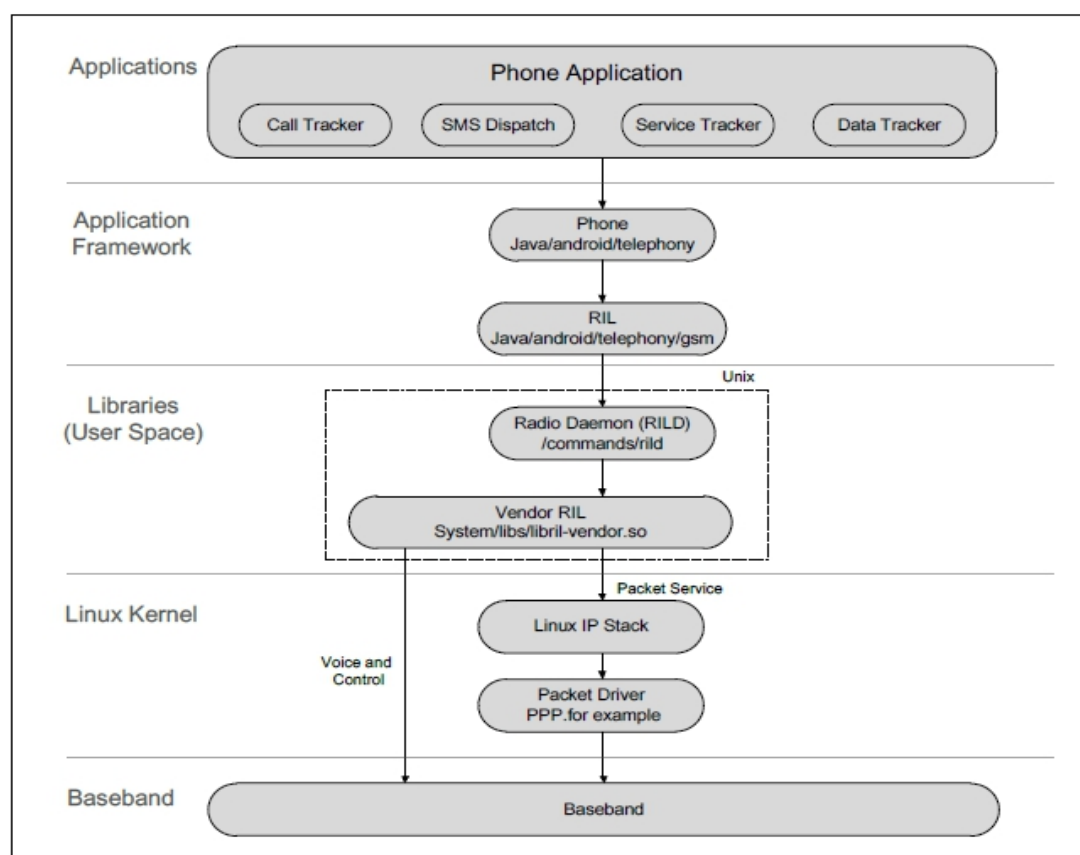
Android7.x RIL 适配说明.....	错误！未定义书签。
目录.....	4
1. 概述.....	5
1.1 RIL 简介.....	
1.2 RIL 提供的功能.....	
1.3 RIL 支持的 android 版本.....	
2.RIL 适配.....	7
2.1. Kernel 驱动配置.....	
2.2. 安装 USB 驱动.....	
2.3. android 端配置.....	
2.4. 权限配置.....	
2.5. 内置 libmeig-ril.so.....	

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。

1.1 RIL简介

Android ril是个硬件抽象层，是android telephony framework 和 modem(SLM3XX模块)的通信纽带。负责将telephony framework的电信业务请求转化对应的at指令，获取相应的电信业务信息，同时将modem主动上报的一些状态返回给android telephony framework，使android 上位机能够正常的做相关的电信业务。Android ril在整个android系统中的作用如下图所示：



可以看到android ril 分2部分，第一部分是rild，主要负责android framework 和 rild进程的socket通信，另外一部分是vendor ril， 主要是负责rild进程和modem的通信。作为SLM3XX的厂商，我们主要是实现vendor ril部分的功能，从而完成vendor ril和SLM3XX模块的电信业务交互。满足android整机电信相关业务要求。

1.2 RIL提供的功能

功能	是否支持
短信	是
语音	是
数据	是
电话本	待验证

1.3 RIL支持的android版本

版本	是否支持
Android 4.x	是
Android 5.x	是
Android 6.0	是
Android 7.x	是
Android 8.x	是
Android 9.x	是
Android 10.x	待验证

本章主要介绍为了使Android设备利用美格的SLM3XX模块进行电信业务，必须对Android设备端进行一些必要的配置，涉及到Android侧和Linux内核侧两部分，本章主要讲述对驱动端的适配。同时客户必须向美格申请获取最新的ril库。

2.1. Kernel驱动配置

Kernel 需要配置 ppp 驱动与 usb 串口驱动，配置方法有两种：

方法一：修改项目对应的 config 文件。

请修改代码 kernel/arch/arm64/configs/*defconfig 中以下配置为 y，如 32 位系统，修改 kernel/arch/arm/configs/*defconfig

```
CONFIG_PPP_DEFLATE=y
```

```
CONFIG_PPP_FILTER=y
```

```
CONFIG_PPP_MULTILINK=y
```

```
CONFIG_PPP_ASYNC=y
```

```
CONFIG_PPP_SYNC_TTY=y
```

```
CONFIG_USB_SERIAL_GENERIC=y
```

```
CONFIG_USB_SERIAL_OPTION=y
```

方法二：使用 menuconfig 进行配置

```
$ cd <your kernel directory>
```

```
$ make menuconfig
```

启用PPP相关配置

```
Device Drivers --->
```

```
  [*] Network device support --->
```

```
    <*> PPP (point-to-point protocol) support
```

```
      <*> PPP Deflate compression
```

```
        [ * ] PPP filtering
```

```
        [ * ] PPP multilink support
```

```
      <*> PPP support for async serial ports
```

```
      <*> PPP support for sync tty ports
```

启用USB Serial相关配置

```
Device Drivers --->
```

```
  [*] USB support --->
```

```
    <*> USB Serial Converter support --->
```

[*] USB Generic Serial Driver

<*> USB drivers for GSM and CDMA modems

<*> USB Quatech Serial Driver for USB 2 devices

2.2. 安装USB驱动

内核中添加 PID和VID, 修改<kernel>/drivers/usb/serial/option.c文件, 加入 VID=0x2dee PID=0x4d41 0x4d42 0x4d43。

```
/* Meig products */
#define MEIG_VENDOR_ID          0x2dee
#define MEIG_PRODUCT_SLM320_1   0x4d41
#define MEIG_PRODUCT_SLM320_2   0x4d42
#define MEIG_PRODUCT_SLM320_3   0x4d43

static const struct usb_device_id option_ids[] = {
    { USB_DEVICE(MEIG_VENDOR_ID, MEIG_PRODUCT_SLM320_1) },
    { USB_DEVICE(MEIG_VENDOR_ID, MEIG_PRODUCT_SLM320_2) },
    { USB_DEVICE(MEIG_VENDOR_ID, MEIG_PRODUCT_SLM320_3) },
}
```

2.3. android 端配置

确认以下定义包含在file_contexts中, 例如: external/sepolicy/file_contexts

```
/dev/ttyUSB[0-9]*      u:object_r:tty_device:s0
/system/bin/rild         u:object_r:rild_exec:s0
/system/socket/rild       u:object_r:rild_socket:s0
/system/socket/rild-debug u:object_r:rild_debug_socket:s0
/system/bin/pppd         u:object_r:ppp_exec:s0
/dev/ppp                 u:object_r:ppp_device:s0
```

确认以下定义包含在ueventd.rc中, 例如: system/core/rootdir/ueventd.rc

```
/dev/ppp                0660   radio    radio
/dev/ttyUSB*            0666   system   system
```

确认以下定义包含在init.rc中, 例如: system/core/rootdir/init.rc

```
service ril-daemon /system/bin/rild -l /system/lib64/libmeig-ril.so
class main
socket rild stream 660 root radio
socket rild-debug stream 660 radio system
user root
group radio cache inet misc audio sdcard_rw log net_admin
```

备注: 添加配置后编译可能报错, 根据提示注释掉重复定义的内容。

2.4. 权限配置

找到对应的rild.te文件，例如：system/sepolicy/rild.te

```
allow rild self:socket create_socket_perms;
allow rild default_prop:property_service set;
allow rild device:chr_file { read write ioctl open getattr };
allow rild kernel:system module_request;
allow rild net_radio_prop:property_service set;
allow rild ppp_device:chr_file { read write ioctl open };
allow rild ppp_exec:file { getattr read execute open execute_no_trans };
allow rild radio_prop:property_service set;
allow rild self:capability { net_admin setuid };
allow rild shell_exec:file { read execute open execute_no_trans };
allow rild sysfs_wake_lock:file { open read write };
allow rild system_file:file execute_no_trans;
allow rild system_prop:property_service set;
allow rild self:capability dac_override;
```

确认以下修改包含在对应的domain.te文件，例如：external/sepolicy/domain.te

```
@@ -260,7 +260,7 @@ neverallow { domain -kernel -init -recovery -uncrypt }
block_device:blk_file { o
# Rather force a relabel to a more specific type.
# init is exempt from this as there are character devices that only it
uses.
# ueventd is exempt from this, as it is managing these devices.
-neverallow { domain -init -ueventd } device:chr_file { open read write };
+neverallow { domain -init -ueventd -rild } device:chr_file { open read
write };

# Limit what domains can mount filesystems or change their mount flags.
# sdcard_type / vfat is exempt as a larger set of domains need
@@ -340,7 +340,7 @@ neverallow * default_android_service:service_manager
add;

# Require that domains explicitly label unknown properties, and do not
allow
# anyone but init to modify unknown properties.
-neverallow { domain -init } default_prop:property_service set;
+neverallow { domain -init -rild } default_prop:property_service set;
neverallow { domain -init } mmc_prop:property_service set;
```

备注：添加权限后编译可能报错，根据提示注释掉重复定义的内容。

2.5. 内置libmeig-ril.so

Vendor ril 部分编译出来的是一个 so 库，需要把它集成到 android rom 里。

将 meig-ril 文件夹拷贝至 vendor/meig/meig-ril/目录下，如没有 vendor 目录，可以新建或放置到任意可以编译到的目录下。

调试过程可以通过 adb push 将 libmeig-ril.so 推到设备的 system/lib 目录下，如果是 64 位的库需要将此库推到设备的 system/lib64 目录下。

为了支持把相关可执行程序与 ril 库编译到系统中，将 device.mk 添加到相应的项目的 mk 文件中。例如：

device/softwinner/tulip-p1/tulip_p1.mk 文件

```
$(call inherit-product-if-exists, vendor/meig/meig-ril/device.mk)
```