MEIG-SLM3XX系列RIL Android9.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_Android9.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

美格智能技术股份有限公司
MEIG SMART TECHNOLOGY CO.,LTD
地址：中国·上海市·徐汇区钦江路88号西座三楼
电话：86-21-54278676 传真：86-21-54278679
邮箱：m2msales@meigsmart.com 网址：www.meigsmart.com

重要声明

版权声明

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	修改文件名称

目录

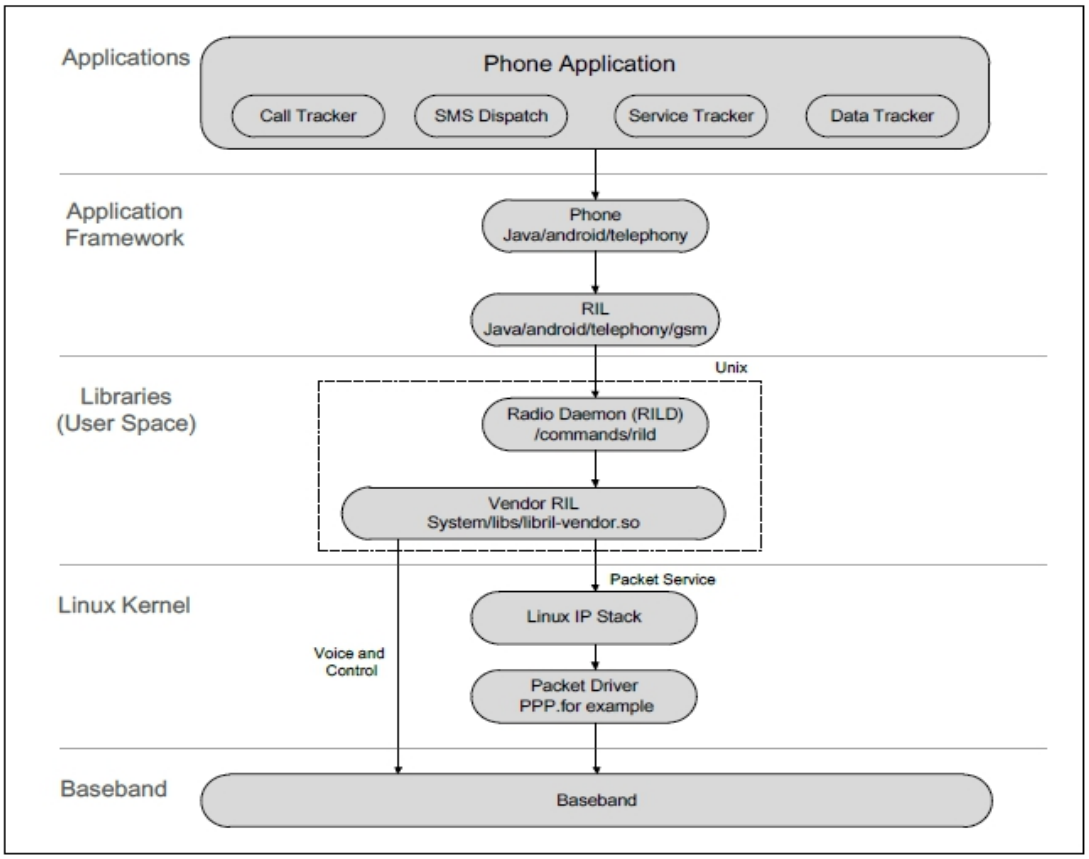
目录

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， 主要是否负责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

2. RIL 适配

本章主要介绍为了能使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：
system/sepolicy/prebuilts/api/28.0/private/file_contexts

```
/dev/ttyUSB[0-9]*           u:object_r:tty_device:s0
```

system/sepolicy/vendor/file_contexts

```
/(vendor|system/vendor)/bin/hw/rild          u:object_r:rild_exec:s0
```

修改对应的system.prop,例如：device/qcom/msm8953_64/system.prop

```
-#rild.libpath=/vendor/lib64/libril-qc-qmi-1.so
-#rild.libargs=-d /dev/smd0
+rild.libpath=/system/lib64/libmeig-ril.so
+rild.libargs=-d /dev/ttyUSB0
```

保证ril-daemon的启动方式如下：

```
service ril-daemon /vendor/bin/hw/rild -l /vendor/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

Android9以上版本，权限更加严格，原生情况rild.te在public domia.te不被识别，这边提供两种方案。
方案1：关闭selinux
修改system/core/init/selinux.cpp

```
bool IsEnforcing() {
+    if(true)return false;
  if (ALLOW_PERMISSIVE_SELINUX) {
      return StatusFromCmdline() == SELINUX_ENFORCING;
  }
```

方案2：添加部分权限并绕过原生android不识别type rild情况

修改文件device/qcom/sepolicy/vendor/common/rild.te。最上面添加
typeattribute rild mlstrustedsubject;

```
allow rild self:capability { dac_override };
allow rild default_prop:property_service{ set };
allow rild ppp_exec:file{ read open execute execute_no_trans getattr };
allow rild ppp_device:chr_file{ open read write ioctl getattr };
allow rild shell_exec:file{ execute read open execute_no_trans getattr };
allow rild rootfs:dir{ read open };
allow rild system_file:file{ execute_no_trans };
allow rild toolbox_exec:file{ read open getattr execute execute_no_trans };
allow rild default_prop:file{ read };
allow rild net_data_file:dir{ read search };
allow rild net_data_file:file{ read open getattr };
allow rild net_radio_prop:file{ read open getattr };
allow rild net_radio_prop:property_service{ set };
```

修改文件system/sepolicy/public/property.te和
system/sepolicy/prebuilts/api/28.0/public/property.te

```
compatible_property_only(`
# Prevent properties from being set
 neverallow {
   domain
   -coredomain
   -appdomain
   -vendor_init
   -mlstrustedsubject
 } {
   core_property_type
   extended_core_property_type
   exported_config_prop
   exported_dalvik_prop
   exported_default_prop
   exported_dumpstate_prop
   exported_ffs_prop
   exported_fingerprint_prop
   exported_system_prop
   exported_system_radio_prop
   exported_vold_prop
   exported2_config_prop
   exported2_default_prop
   exported2_system_prop
   exported2_vold_prop
   exported3_default_prop
   exported3_system_prop
   -nfc_prop
   -powerctl_prop
   -radio_prop
 }:property_service set;
```

```
neverallow {
  domain
  -coredomain
  -appdomain
  -hal_nfc_server
  -mlstrustedsubject
} {
  nfc_prop
}:property_service set;
```

```
# Prevent properties from being read
  neverallow {
    domain
    -coredomain
    -appdomain
    -vendor_init
    -mlstrustedsubject
  } {
    core_property_type
    extended_core_property_type
    exported_dalvik_prop
    exported_ffs_prop
    exported_system_radio_prop
    exported2_config_prop
    exported2_system_prop
    exported2_vold_prop
    exported3_default_prop
    exported3_system_prop
    -debug_prop
    -logd_prop
    -nfc_prop
    -powerctl_prop
    -radio_prop
  }:file no_rw_file_perms;
```

修改文件 system/sepolicy/public/domain.te和
system/sepolicy/prebuilts/api/28.0/public/domain.te

```
  neverallow { domain -init -vendor_init -mlstrustedsubject }
  default_prop:property_service set;
  neverallow { domain -init -vendor_init } mmc_prop:property_service set;

  compatible_property_only(`
      neverallow { domain -init -mlstrustedsubject }
  default_prop:property_service set;
      neverallow { domain -init } mmc_prop:property_service set;
      neverallow { domain -init -vendor_init }
  exported_default_prop:property_service set;
      neverallow { domain -init } exported_secure_prop:property_service set;
  ')
```

```
  # vendor domains may only access files in /data/vendor, never
core_data_file_types
  neverallow {
    domain
    -appdomain # TODO(b/34980020) remove exemption for appdomain
    -coredomain
    -data_between_core_and_vendor_violators # TODO(b/34980020) Remove once
all violators have been cleaned up
    -vendor_init
    -mlstrustedsubject
  } {
    core_data_file_type
    # libc includes functions like mktime and localtime which attempt to
access
    # files in /data/misc/zoneinfo/tzdata file. These functions are
considered
    # vndk-stable and thus must be allowed for all processes.
    -zoneinfo_data_file
  }:file_class_set ~{ append getattr ioctl read write map };
```

```
  # vendor domains may only access dirs in /data/vendor, never
core_data_file_types
  neverallow {
    domain
    -appdomain # TODO(b/34980020) remove exemption for appdomain
    -coredomain
    -data_between_core_and_vendor_violators
    -vendor_init
    -mlstrustedsubject
  } {
    core_data_file_type
    -system_data_file # default label for files on /data. Covered below...
    -vendor_data_file
    -zoneinfo_data_file
  }:dir *;
```

```
full_treble_only(`
    # Do not allow vendor components to execute files from system
    # except for the ones whitelist here.
    neverallow {
        domain
        -coredomain
        -appdomain
        -vendor_executes_system_violators
        -vendor_init
        -mlstrustedsubject
    } {
        exec_type
        -vendor_file_type
        -crash_dump_exec
        -netutils_wrapper_exec
        -system_file
        -ppp_exec
        -shell_exec
    }:file { entrypoint execute execute_no_trans };
')
```

```
neverallow {
  domain
  -dnsmasq
  -dumpstate
  -init
  -installd
  -install_recovery
  -lmkd
  -netd
  -perfprofd
  -postinstall_dexopt
  -recovery
  -sdcardd
  -tee
  -ueventd
  -uncrypt
  -vendor_init
  -vold
  -vold_prepare_subdirs
  -zygote
  -mlstrustedsubject
} self:capability dac_override;
```

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)
```