



联发科安卓 SDK 开发者指南

版本: **1.0**

发布日期: **2014 年 9 月 8 日**

规范如有变更恕不另行通知。

© 2014 MediaTek Inc.

本文件包含的内容为联发科技有限公司所有。
严禁未经授权擅自复制全部或部分本文信息。

目录

1	介绍	5
2	联发科 Android 手机技术.....	6
2.1	HotKnot 无线数据传输	6
2.2	多 SIM 卡	7
2.3	多媒体操作.....	8
2.3.1	肖像增强功能	8
2.3.2	图像变换	8
2.3.3	高清录音和文件标签	9
2.4	多媒体捕捉.....	9
2.4.1	手势检测	9
2.4.2	多视角（MAV）图像捕捉	9
2.4.3	摄像机音频和视频设置	10
3	联发科设备的开发.....	11
3.1	联发科 Android SDK	11
3.2	Android 参考手机	11
4	入门	12
4.1	系统必备组件	12
4.2	下载	12
4.3	安装	13
4.3.1	安装 API 包	13
4.3.2	安装工具程序包	13
4.4	使用联发科 API 进行开发	14
4.4.1	联发科功能的 API	14
4.5	联发科兼容性 API	15
4.6	配置和使用联发科模拟器	18
4.6.1	替换标准的模拟器	18
4.6.2	创建一个 Android 虚拟设备	19
4.6.3	其他配置和工具	22
4.7	联发科模拟器支持的 API	23

4.8	联发科 DDMS 指南	23
4.8.1	运行联发科 DDMS	23
4.8.2	模拟来自于任一 SIM 卡的电话	24
4.8.3	模拟从任一 SIM 卡中发出的短信	24
4.8.4	模拟耳机设备的变化	25
4.8.5	模拟电源状态	26
4.8.6	模拟传感器状态	27
4.8.7	模拟 SD 卡状态	27
4.9	其他工具	28
4.9.1	Intel® 硬件加速执行管理器	28
5	API 指南	29
5.1	API 级别和兼容性	29
5.2	使用 HotKnot (com.mediatek.hotknot)	30
5.2.1	创建 HotKnot 发送应用程序	30
5.2.2	创建 HotKnot 接收应用程序	31
5.3	使用多 SIM (com.mediatek.telephony)	32
5.3.1	TelephonyManagerEx 类	32
5.3.2	SmsManagerEx 类	32
5.4	使用 MAV 捕捉 (com.mediatek.hardware)	34
5.4.1	CameraEx 类	34

图表目录

表 1 AVD 设置的建议值	21
表 2 联发科为 SDK 的 API 支持	23
表 3 按级别所示的联发科 API 支持	30
图 1 屏幕间的 TX 和 RX 网络之间的相互作用	6
图 2 处理前（左）和肖像的增强处理之后（右）	8
图 3 解压 API ZIP 的内容到 Android SDK	13
图 4 用联发科 API 编码	14
图 5 为兼容性 API 创建 libs 文件夹	15
图 6 复制兼容性 API jar 到项目中	15
图 7 应用程序项目中兼容性 API JAR 文件	16
图 8 解压的兼容性 API JAR 显示出可用的 API	16
图 9 在 bin 文件夹中生成的项目 APK 文件	17
图 10 用联发科模拟器文件替换标准模拟器文件	18
图 11 Android 虚拟设备管理器	19
图 12 联发科设置模拟 AVD 设置	20
图 13 启动联发科模拟器	21
图 14 联发科模拟器显示双 SIM 卡图标	22
图 15 使用 Call1 和 Call2 模拟通话。	24
图 16 使用 Send1 和 Send2 模拟发送信息。	24
图 17 各种耳机设备的模拟	25
图 18 各种电源的模拟	26
图 19 传感器的各种状态	27
图 20 模拟插入或取出 SD 卡的选项	27
图 21 Windows 下查询 HAXM 是否存在	28
图 22 Linux 下查询 HAXM 是否存在	28
图 23 跨级别的 API 支持	29

1 介绍

本开发者指南介绍了联发科的 **Android SDK**。它提供了安装和使用 **SDK** 的所有信息，并介绍基于联发科芯片的 **Android** 设备的独特技术信息。

在联发科技 创意实验室网站上对 **SDK** 详细的 **API** 文档提供了补充 [API 参考](#)。

2 联发科 Android 手机技术

联发科芯片在[不断增长的 Android 设备家族](#)中占据核心位置。我们的芯片为 Android 手机带来多 SIM 卡、文件传输（使用 HotKnot）、成像、视频和手势控制功能。本章将为您介绍这些技术。

2.1 HotKnot 无线数据传输

HotKnot 是联发科发明的一种创新的数据和文件传输机制。它利用了电容式触摸屏的物理特性，使数据在两个设备之间进行交换。

该技术利用了电容式触摸屏的特性，使得两个手机靠近时（小于图 1 厘米），一个手机屏幕的传输网格可以与另一个手机的接收网格进行交互。这个过程如图 1 所示。

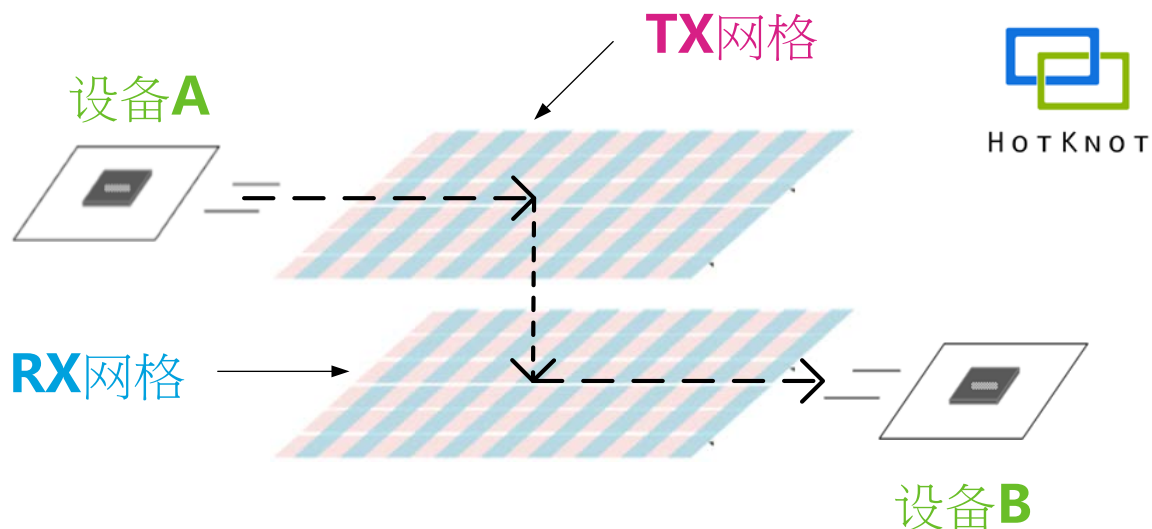


图 1 屏幕间的 TX 和 RX 网格之间的相互作用

这种技术对于设备制造商的吸引力在于，不像近场通讯（NFC）需要增加无线电功能和在 BOM 中添加天线，HotKnot 是利用手机屏幕工作的，该组件已经在他们的 BOM 表中。

HotKnot 为用户提供类似 NFC 的点击发送/共享数据交换功能。然而，不像 NFC，当设备在非常近距离时被窥探或黑客入侵的危险要小得多。

HotKnot（[com.mediatek.HotKnot](#)）的联发科 API 使您能充分利用该技术，通过检测启用手机触摸屏的 HotKnot 的存在，建立两个手机之间的连接，以及手机之间的数据传输。

用 HotKnot 可以做 NFC 能做的手机到手机数据传输的任何事情，例如：

- 交换网站地址，联系人数据，照片和视频
- 交换游戏进程
- 交换应用程序之间的离散数据
- 交换设备性能数据，蓝牙配对，WiFi 连接，或其他服务连接
- 便利移动支付

2.2 多 SIM 卡

多 SIM 技术使设备使用 2 个或多个 SIM 卡来访问移动网络。这使用户能够从一个电话使用多个移动用户。典型的例子可以包括：

- 一个 SIM 用作个人用户，另一个用于企业或雇主的用户。
- 经常旅行的人可能有一个 SIM 卡用于自己的本国网络，另一个用于他们要前往的国家的当地网络。
- 利用不同运营商之间不同的包月套餐；一个 SIM 卡可以提供便宜的电话，另一个用于便宜的数据流量。

联发科的 API (com.mediatek.telephony)，使您可以获取可用的 SIM/用户的信息，并侦听 SIM 卡变化或活动，如呼叫或使用数据流量，或所连接电池的详细信息。此外，该 API 允许应用程序创建短信，并使用特定 SIM 卡发送。

2.3 多媒体操作

利用硬件优化，联发科提供给您大量的图像和音频处理增强功能，可以为您提供性能改进和功能独特的应用程序。

2.3.1 肖像增强功能

就像在商业工作室使用的为杂志和广告增强肖像的工具，这些图像效果使您可以创建应用为用户提供增强他们捕捉任何肖像风格的能力。这些效果提供的范围包括：

- 皮肤平滑
- 肤色调整
- 脸部“减肥”
- 眼睛放大



图 2 处理前（左）和肖像的增强处理之后（右）

图像编辑效果（com.mediatek.effect）API 使您能够在图像编辑应用程序中充分利用这些功能。

2.3.2 图像变换

联发科芯片组对许多图像编辑工具提供带有硬件加速的支持，可以：

- 水平或垂直翻转图像
- 顺时针旋转 90、180 或 270 度
- 裁剪图像
- 对图像进行抖动处理
- 调整 JPEG 图像的编码质量
- 调整图像的锐度

这些 API（com.mediatek.imagetransform）使您能够充分利用这些功能，并提供更敏感的图像编辑应用程序。

2.3.3 高清录音和文件标签

联发科 Android 芯片组包含高清录音和标记音频文件的功能。

高清音频为用户提供了更好的录音品质。该技术应用于由降噪、动态范围压缩和立体声增强技术支持的 48KHz 立体声录音。

在您的音频应用程序中，该 API 可使高清录音模式进行设定，录制暂停和重新启动。高清录音模式调节应用于以下录音的降噪水平：

- 正常（降噪关闭），用于需要捕捉环境声学特性的场合，例如录制音乐时。
- 室内（中等水平降噪），用于录制接近主声源的场合，例如录制会议或声音记录时。
- 户外（低水平降噪），用于有较高的背景噪声，或声源在一定距离的情况下，如一个演讲会的发言者以及在户外录音。

此外，该 API 提供了设置专辑和艺术家信息的音频文件标签。这意味着，从您的应用程序录制的声音在设备的音乐播放器中保持正常指数，例如，用户可以创建一个“杰克的生日”专辑，录制“祝杰克生日快乐”，“杰克打开他的礼物”，或更多，然后即时从他们的音乐播放器访问录音。

这些 API（[com.mediatek.media](#)）使您在音频应用程序中充分利用高清录音设置和文件标签。

2.4 多媒体捕捉

2.4.1 手势检测

配置联发科芯片的设备为应用提供观看设备的相机图像流，检测张开的手掌和胜利（V）手势的能力。

使用此功能的应用程序可以启动基于这些手势操作，如拍照，启动或停止一个过程。例如，一个计时的应用程序会在张开手掌时启动秒表开始计时，当用户表示已经完成胜利的手势时停止。

这些 API（[com.mediatek.gesture](#)）使您在应用程序中监视和检测手势。

2.4.2 多视角（MAV）图像捕捉

使用这种技术用户可以拍摄整个物体或人物。这些记录的图片存储在 [Camera and Imaging Products Association Multi-Picture Format](#) 文件中，以及每个图像捕捉的角度（由设备的加速计提供）的信息。

应用程序使用这些图像可以创建 3D 模型或拍摄的物体或人物的其他格式，这开辟了利用新兴技术如 3D 打印的可能性。

这些 API（[com.mediatek.hardware](#)）使您能够添加 MAV 文件到您的应用程序。然后，您可以对添加的图形文件进行额外处理。

2.4.3 摄像机音频和视频设置

此功能使您的应用程序可以访问前面的和后面的摄像机设置细节，包括：

- 视频：输出文件的视频编解码格式、比特率、帧速率和分辨率。
- 音频：音频编解码格式、比特率、采样率和用于录音的音频信道数目。

这些 API（com.mediatek.camcorder）使您能够在您的视频和音频应用程序中充分利用高清拍摄设置和文件标签。

3 联发科设备的开发

本章概述了使用联发科芯片的 Android 设备功能的应用程序的开发和测试。

3.1 联发科 Android SDK

为了能够开发应用程序，联发科的 Android SDK 是基于搭载联发科芯片的 Android 设备的参考实现的。它提供一组对于谷歌 Android SDK 的扩展。该 SDK 提供：

- 您将导入您代码中的联发科的 API 库。
- 联发科的 x86 模拟器，它提供功能来测试您的应用程序的电话和多 SIM 卡功能。除了所有常见的 Android 模拟器功能，模拟器提供了一个振动指示器和可热插拔的 SD 卡。
- 定制 Dalvik 调试监视器服务器（DDMS），增加了测试电话功能，如插、拔耳机。联发科 DDMS 也可以用来发送语音命令来测试呼叫和短信功能。

3.2 Android 参考手机

联发科模拟器提供对语音通信和多 SIM 卡功能的测试。对于利用了联发科功能的应用程序的所有测试都将要在设备上完成。适合测试的设备的最新列表在联发科技创意实验室 网站上的 [Android 参考手机](#) 页面中提供。

4 入门

这一章为您提供了安装、配置和使用联发科 Android SDK 的详细指南。



本章提供的说明是基于在 **Microsoft Windows** 上安装和使用 SDK。除非另有说明，这些说明也适用于使用苹果 **Mac** 和 **Linux** 电脑的 SDK。



联发科模拟器和定制的 DDMS 目前不适用于苹果 **Mac**。

4.1 系统必备组件

在安装联发科的 Android SDK 之前，你应先安装了 Android SDK，包括所需的支持软件。更多信息请参阅 Android Developer 网站的 [Get the Android SDK](#)。

4.2 下载

最新版本的联发科 Android SDK 包可以从联发科技创意实验室网站得到。软件包包括：

- **mediatek_android_sdk_api-<api_package_version>.zip** — 联发科 API 包。包含用于 Microsoft Windows 和 Linux 的联发科 SDK API 库和联发科模拟器组件。
- **mediatek_android_sdk_tools-<tools_package_version>.zip** — 联发科工具包。包含用于 Microsoft Windows 和 Linux 的 DDMS 定制版本。
- **Release-<sdk_version>.txt** — 版本摘要将提供在此版本中所做的更改说明。

4.3 安装

本节介绍安装 SDK 和工具包的步骤。

4.3.1 安装 API 包

安装联发科 Android SDK API 包需要：

- 1 把 mediatek_android_sdk_api-<api_package_version>.zip 文件的内容解压到您电脑上 Android sdk/add-ons 文件夹中的一个新的文件夹内，如 sdk\add-ons\mtk_sdk_api_addon-19.3，如图 3 所示。

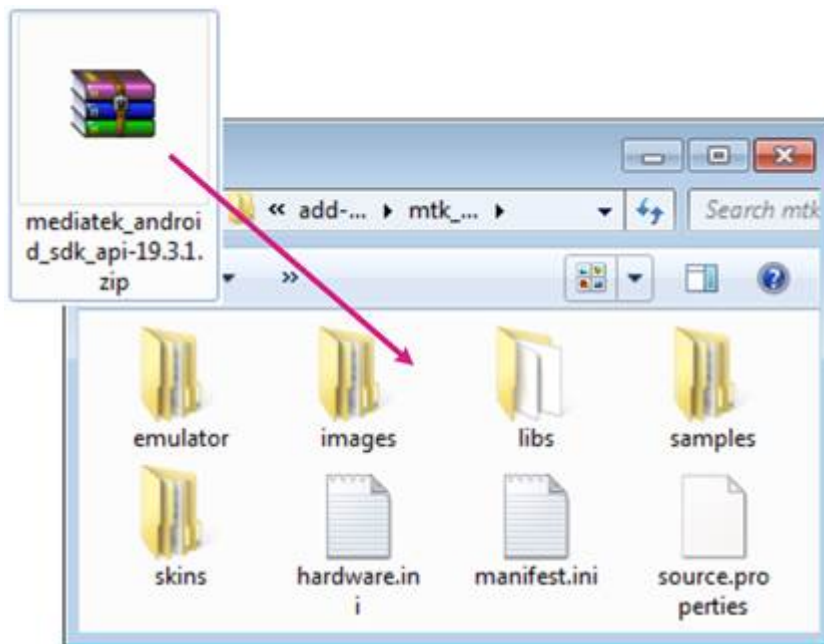


图 3 解压 API ZIP 的内容到 Android SDK

4.3.2 安装工具程序包

安装联发科 Android SDK 工具包（定制的 DDMS）需要：

- 1 将 mediatek_android_sdk_tools<tools_package_version>.zip 解压到本地磁盘的一个方便的文件夹中。

4.4 使用联发科 API 进行开发

此时，您可以开始用联发科 API 编写代码了。

4.4.1 联发科功能的 API

选择 **MediaTek Add-On (MediaTek Inc.)** 作为项目目标。然后您就可以处理 API，如图 4 所示。

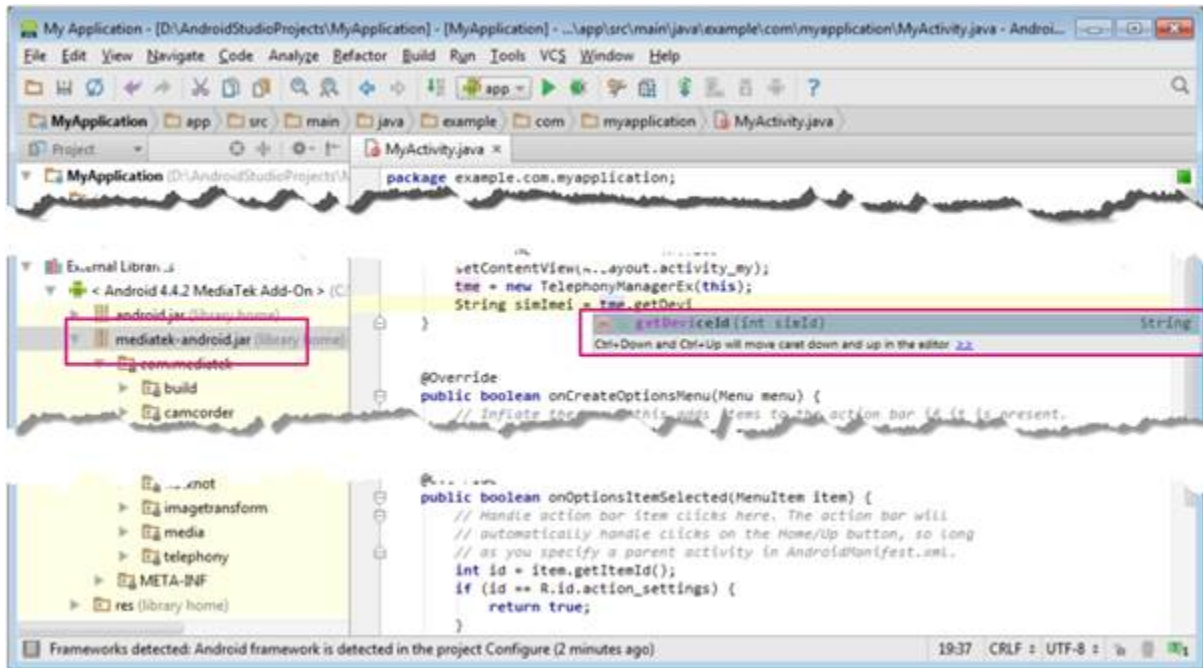


图 4 用联发科 API 编码

4.5 联发科兼容性 API

当您希望您的应用既能应用在联发科芯片的 Android 设备上，又能在其它芯片的 Android 设备上运行，兼容性 API 可以使您的应用程序来测试联发科 API 的存在。使用这个 API 需要进行以下操作：

- 1 在您的项目中创建一个 libs 文件夹，如图 5 所示。

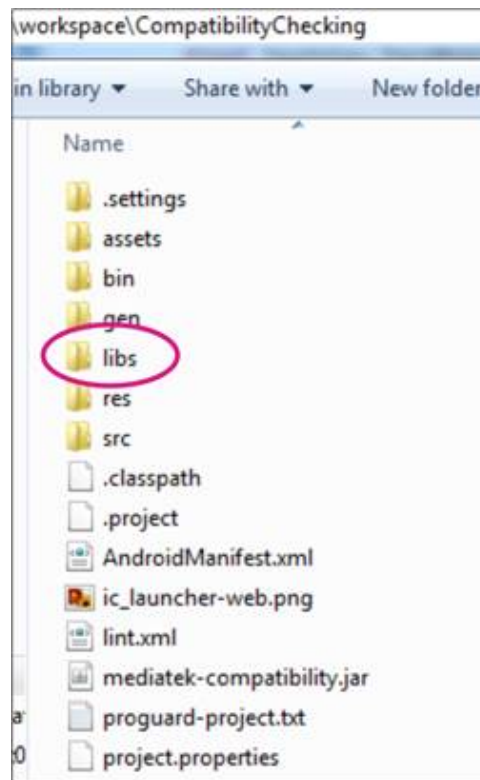


图5 为兼容性API 创建libs 文件夹

- 2 从 Android SDK 插件文件夹 mediatek_android_sdk_api-
<api_package_version>\libs 复制 mediatek-compatibility.jar 到您的项目中 libs
文件夹内,如图 6 所示。

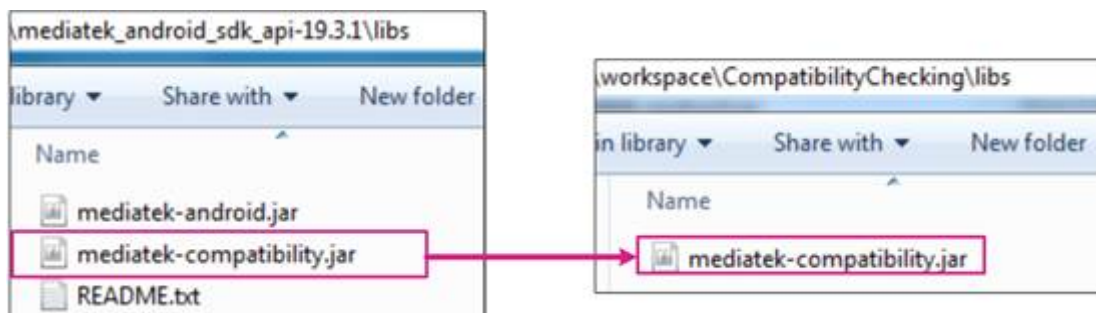


图 6 复制兼容性 API jar 到项目中

- 3 刷新该项目查看 JAR 文件是否存放在 libs 文件夹，参见图 7。

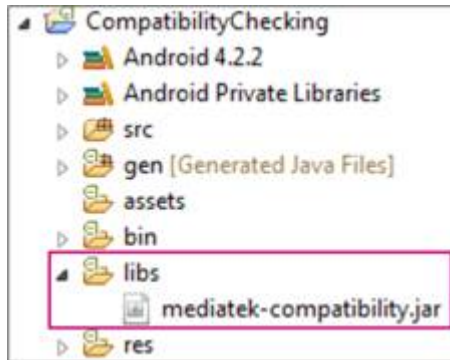


图 7 应用程序项目中兼容性 API JAR 文件

- 4 您现在可以展开 mediatek-compatibility.jar 来查看 API，如图 8 所示。

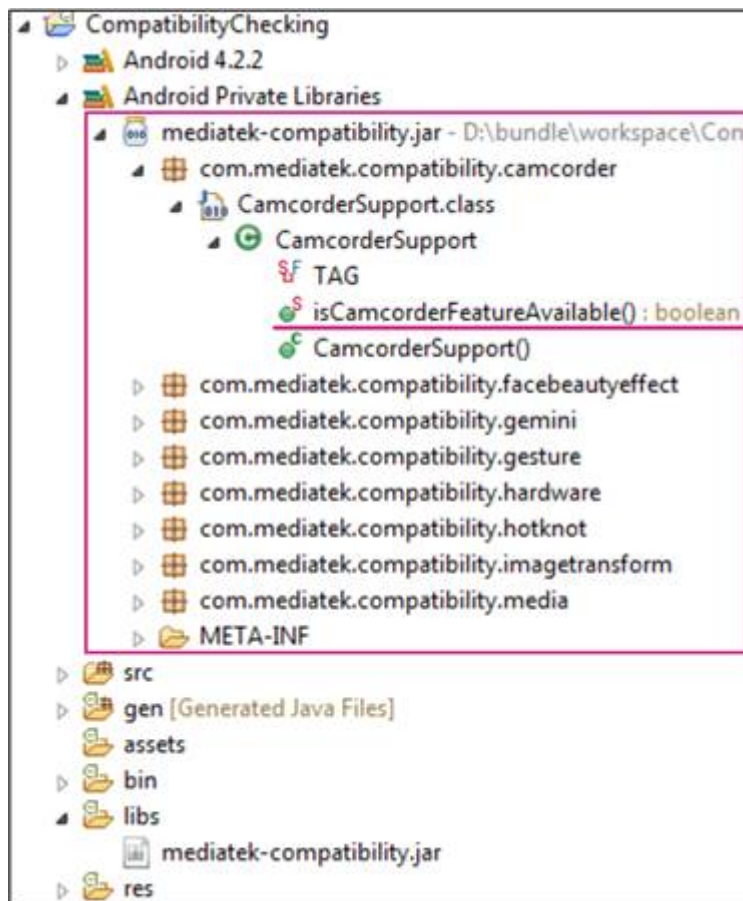


图 8 解压的兼容性 API JAR 显示出可用的 API

- 5 使用 API 编写兼容性检查代码，例如：

```
If (com.mediatek.compatibility.camcorder.CamcorderSupport.\
    isCamcorderFeatureAvailable(this)==true){
    // body
}
```


6 编译项目以创建应用程序 APK 文件，通常将其放在 bin 文件夹中，如图 9 所示。

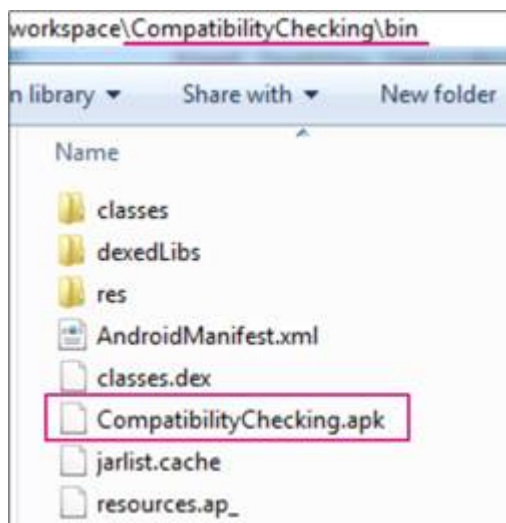


图 9 在 bin 文件夹中生成的项目 APK 文件

4.6 配置和使用联发科模拟器

本节说明如何配置联发科模拟器，并用它来测试您的应用程序的电话和多 SIM 卡功能。



联发科模拟器和定制 DDMS 目前不适用于苹果 Mac。

4.6.1 替换标准的模拟器

使用联发科模拟器的第一步是替换标准的 Android 模拟器可执行文件。



你可以备份一下原来的模拟器文件，以便以后可以恢复它们。

- 1 从 android-sdk/add-ons/mtk_sdk_addon-<tools_package_version>/emulator/<OS version> 复制联发科模拟器文件到 android-sdk/tools 中，替换对应的文件，如图 10 所示。

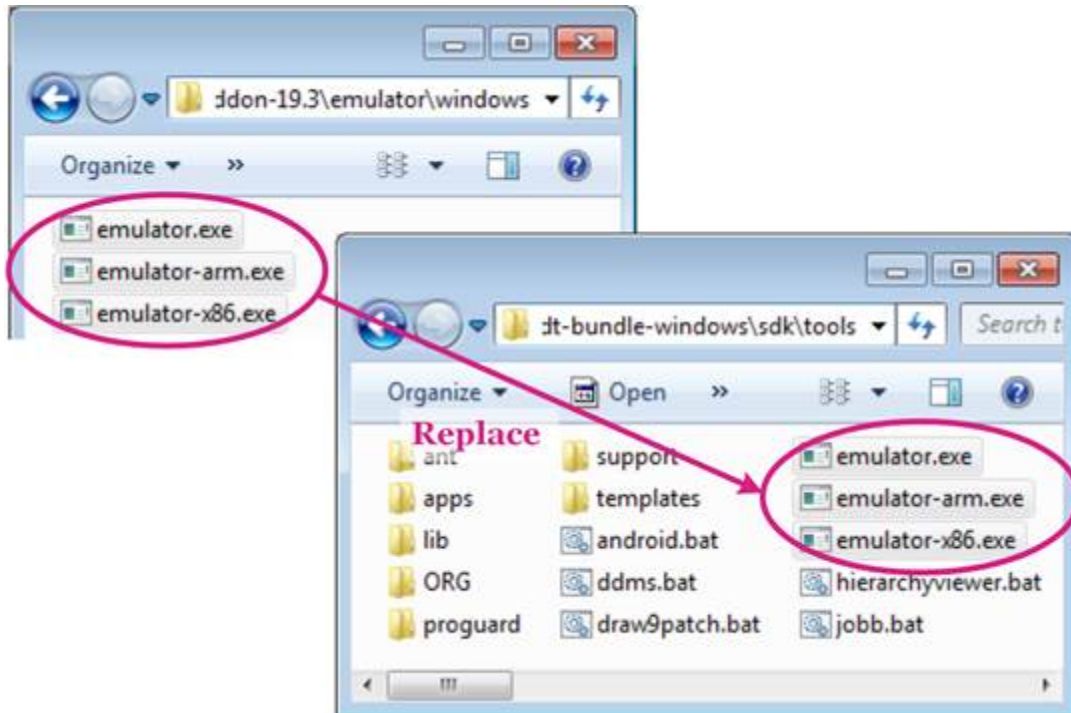


图 10 用联发科模拟器文件替换标准模拟器文件

4.6.2 创建一个 Android 虚拟设备

为了使 Eclipse IDE 能够在联发科模拟器运行并调试软件，你首先需要创建一个 Android 虚拟设备（AVD）。AVD 通过定义将被模拟的硬件和软件选项提供了设备的模型。在 Eclipse IDE 中的设置如下所示

- 1 在 **Window** 菜单上点击 **AVD Manager**。
- 2 在 Android 虚拟设备管理器中，单击 **Create**。如图 11 所示

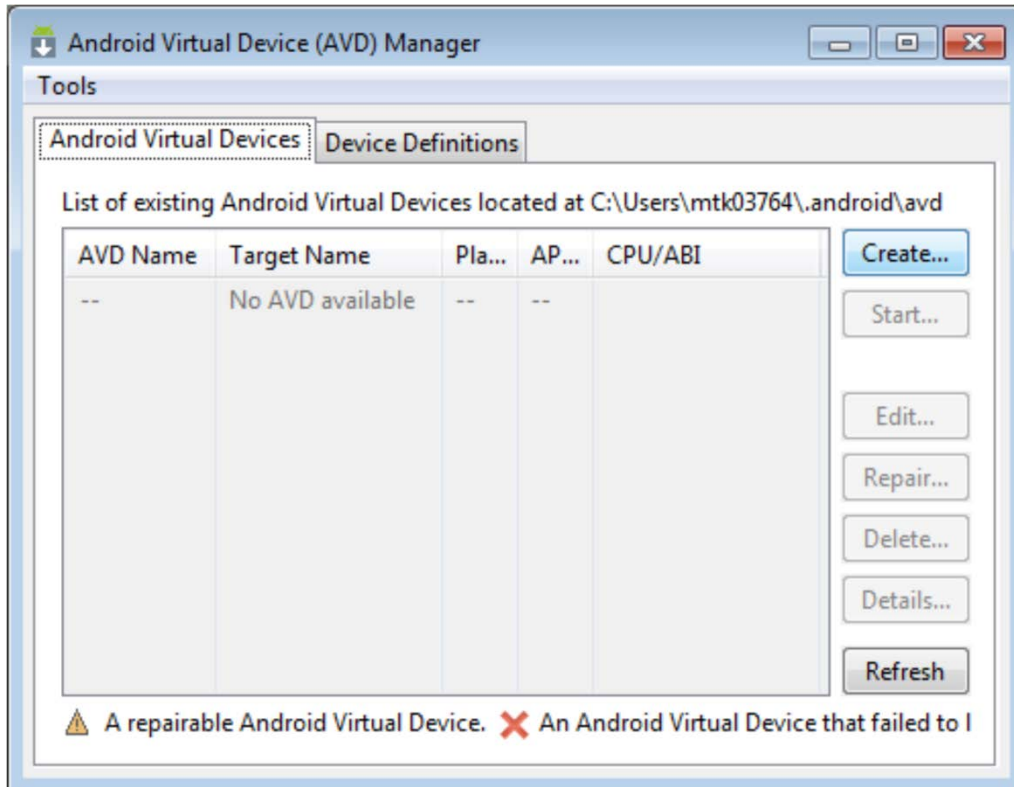


图 11 Android 虚拟设备管理器

- 3 在 **Create new Android Virtual Device (AVD)** 对话框（如图 12 所示），**AVD Name**, **Device**, **Target**, 以及其他功能 - 表 1 中提供的推荐值。

The screenshot shows the 'Create new Android Virtual Device (AVD)' dialog box with the following settings:

- AVD Name:** MTK_720
- Device:** Galaxy Nexus (4.65", 720 × 1280: xhdpi)
- Target:** MediaTek Add-On (MediaTek Inc.) - API Level 19
- CPU/ABI:** Intel Atom (x86)
- Keyboard:** ☒ Hardware keyboard present
- Skin:** MTK720
- Front Camera:** Emulated
- Back Camera:** Emulated
- Memory Options:** RAM: 768, VM Heap: 64
- Internal Storage:** 200 MiB
- SD Card:**
 - ☒ Size: 1024 MiB
 - ☐ File: [Browse...]
- Emulation Options:** ☐ Snapshot, ☒ Use Host GPU
- ☐ Override the existing AVD with the same name

Buttons: OK, Cancel

图 12 联发科设置模拟 AVD 设置

	Windows 7	Windows 7	Linux Ubuntu	Linux Ubuntu
分辨率	720	1080	720	1080
设备	Galaxy Nexus (4.65", 720x1280)	Nexus 5 (4.95",1080x1920)	Galaxy Nexus (4.65", 720x1280)	Nexus 5 (4.95",1080x1920)
目标	MediaTek Add-On (MediaTek Inc.) - API 级别 19	MediaTek Add-On (MediaTek Inc.) - API 级别 19	MediaTek Add-On (MediaTek Inc.) - API 级别 19	MediaTek Add-On (MediaTek Inc.) - API 级别 19
CPU/ABI	Intel Atom (x86)	Intel Atom (x86)	Intel Atom (x86)	Intel Atom (x86)
键盘	Yes	Yes	Yes	Yes
皮肤	MTK720	MTK1080	MTK720	MTK1080
前置摄像头	仿真	仿真	仿真	仿真
后置摄像头	仿真	仿真	仿真	仿真
RAM	512	768	1024	2048
VM 堆	64	64	64	64
内存	200	200	200	200
SD 卡	256	256	256	256
快照	否	否	否	否
使用主机 GPU	是	是	是	是

表 1 AVD 设置的建议值

- 当您输入 AVD 设置后，单击 **OK**（确定）。
- 在 Android 虚拟设备管理器（AVD）中，单击 **Start** 启动联发科模拟器。

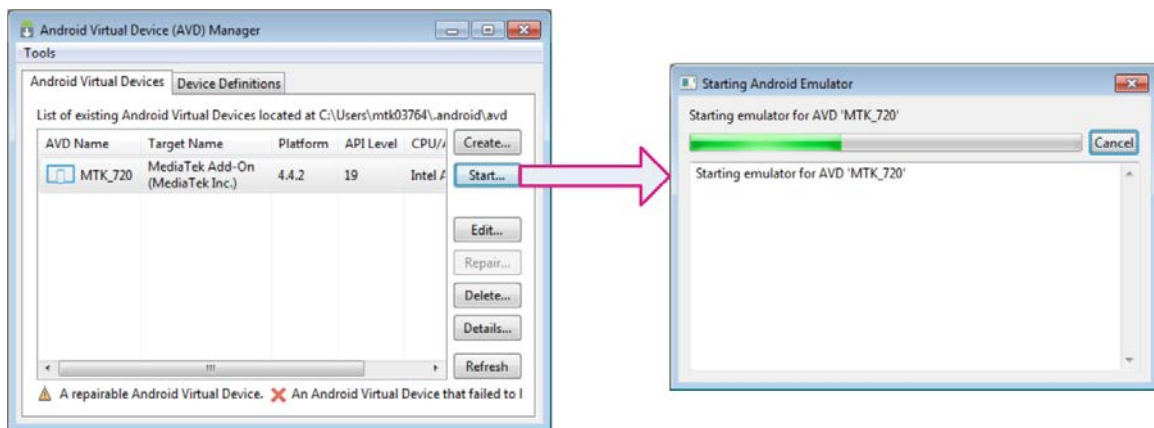


图 13 启动联发科模拟器

- 6 模拟器启动后，在状态栏中会出现两个 SIM 卡的图标。如果看不到图标，请关闭模拟器，按照安装步骤 2 中所述检查您是否已替换模拟器的可执行文件，完成后重新启动模拟器。



图 14 联发科模拟器显示双 SIM 卡图标

此时，模拟器已经设置完成。



模拟器是基于 Android SDK 工具第 17 版以及对 GPU 实验支持的主机电脑上创建的。请参阅 Android 开发网站上有关[配置图形加速器](#)的注释。

4.6.3 其他配置和工具

您会发现，模拟器性能可以通过使用 Intel®HAXM 得到提高。详细信息请参阅有关[Intel® Hardware Accelerated Execution Manager](#)的章节。

4.7 联发科模拟器支持的 API

联发科模拟器不是对所有可用 API 的测试都提供支持; 仅有 SDK 级别版本检查和多 SIM（电话/短信服务）API 可在联发科模拟器进行测试，如表 2 所示。

名称	类别	SDK 级别	是否可在模拟器上运行
MediaTek SDK	SdkVersion	1	是
电话	TelephonyManagerEx	1	是
短信	SmsManagerEx	1	是
相机 MAV	CameraEx	1	否
音频/视频	MediaRecorderEx	1	否
	CamcorderProfileEx		
HotKnot	HotKnotAdapter	2	否
	HotKnotMessage		
手势	Gesture	3	否
ImageTransform	ImageTransformFactory	3	否
Portrait enhancements	Effect	3	否
	EffectFactory		

表 2 联发科为 SDK 的 API 支持

4.8 联发科 DDMS 指南

联发科的 Android SDK 中提供了 Dalvik Debug Monitor Server (DDMS)的自定义版本，自定义为以下提供了模拟：

- 来自 SIM1 或 SIM2 的来电
- 来自 SIM1 或 SIM2 的短信
- 耳机设备的变化
- 电源状态
- 传感器值
- SD 卡的插入和取出

本节说明如何运行 DDMS 和利用它的特性来控制联发科模拟器的各个方面。

4.8.1 运行联发科 DDMS

启动自定义 DDMS 需打开从它安装文件夹中的 ddms.jar 文件。

4.8.2 模拟来自于任一 SIM 卡的电话

为了模拟从 SIM1 和 SIM2 的来电，在 **Emulator Controls** 中点击 **Call1** 或 **Call2**，如图 15 所示。

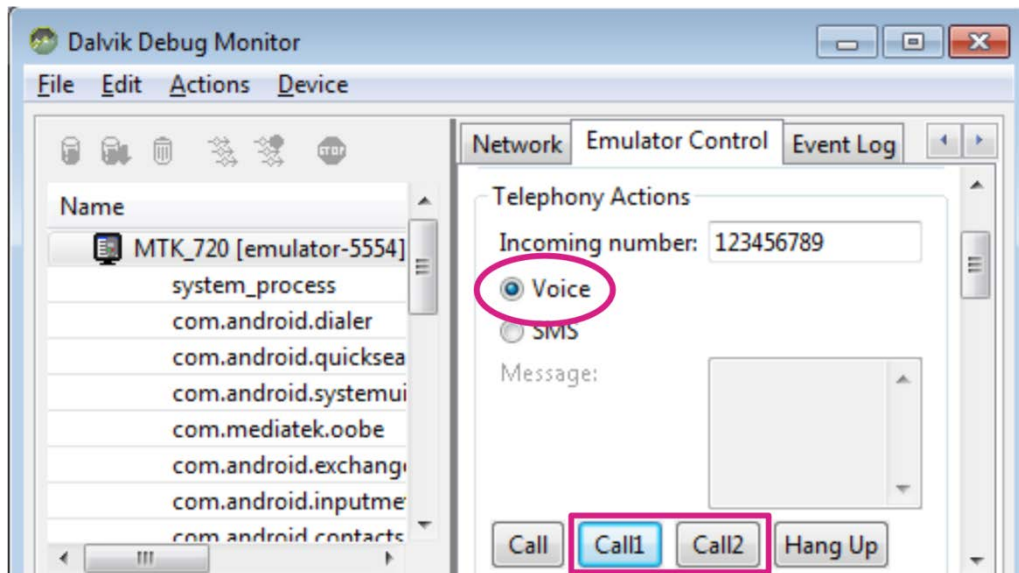


图 15 使用 Call1 和 Call2 模拟通话。

4.8.3 模拟从任一 SIM 卡中发出的短信

为了模拟从 SIM1 和 SIM2 的短信，在 **Emulator Controls** 中点击 **Send1** 或 **Send2**，如图 16 所示。

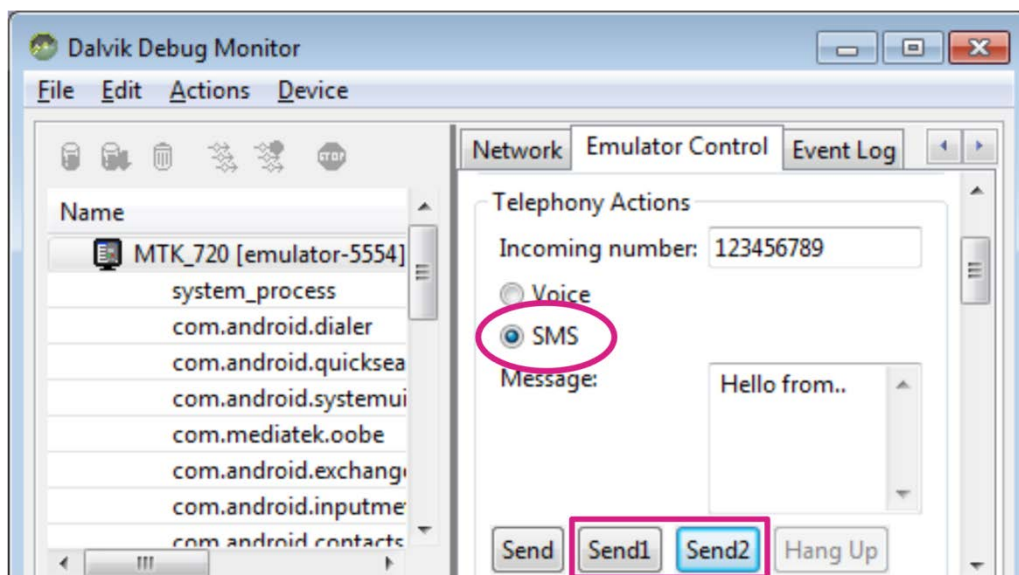


图 16 使用 Send1 和 Send2 模拟发送信息。

4.8.4 模拟耳机设备的变化

如图 17 所示，在 **Emulator Controls** 中 **Headset** 下的耳机设备状态是可以被模拟的：

- 点击 **Unplug headset**，移除耳机。
- 点击 **Plug headset**，加入耳机。
- 点击 **Plug headset (with microphone)**，加入耳机以及麦克风。

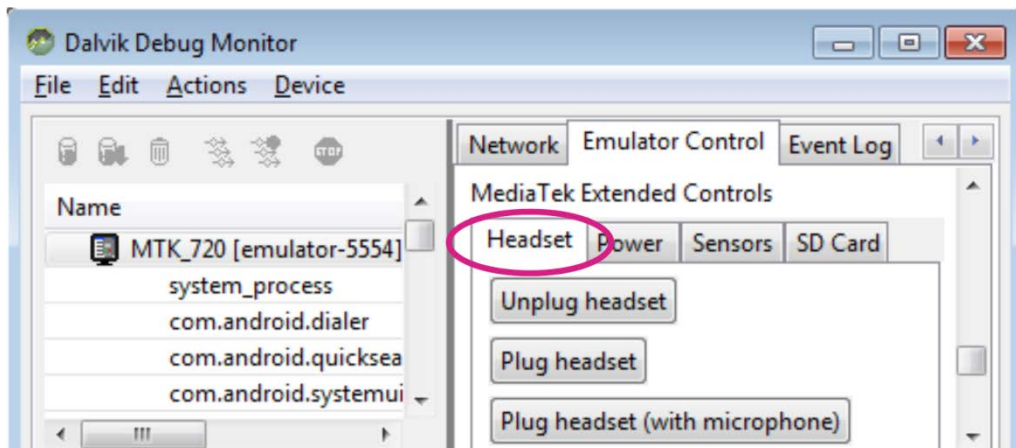


图 17 各种耳机设备的模拟

4.8.5 模拟电源状态

如图 18 所示，在 **Emulator Controls** 中的 **Power** 项下，提供了可用来模拟以下功能的工具：

- 点击 **Set**，设定模拟器中电池 **Capacity** 的级别。
- 点击 **Unknown**，**Charging**，**Discharging**，**Not charging**，或 **Full** 来模拟电池的充电状态。
- 点击 **Unknown**，**Good**，**Overheat**，**Dead**，**Over Voltage**，或 **Unspecified Failure** 来模拟电池的健康状态。
- 点击 **Online** 或 **Offline** 来模拟是否连接充电器。

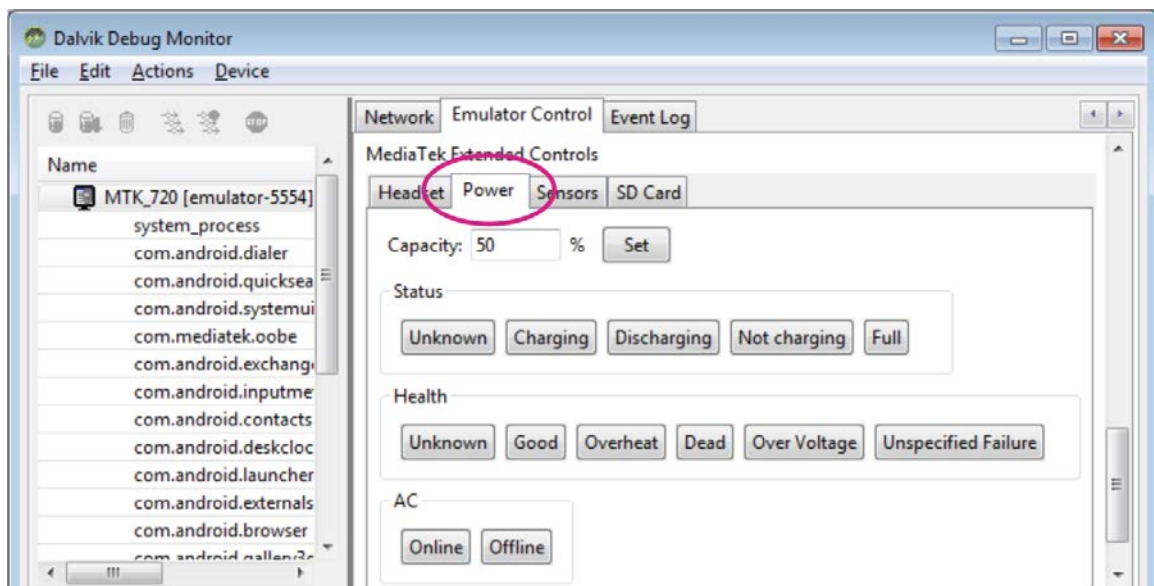


图 18 各种电源的模拟

4.8.6 模拟传感器状态

如图 19 所示，在 **Emulator Controls** 中的 **Sensor** 项下，可以设置以下传感器的值：

- Accelerometer: **X**, **Y**, 和 **Z**。
- Orientation Sensor: **Azimuth**, **Pitch**, 和 **Roll**。
- Magnetic Field Sensor: **X**, **Y**, 和 **Z**。
- Temperature Sensor: temperature.

输入数值后，点击“Set”，模拟器中的传感器值会同步更新。

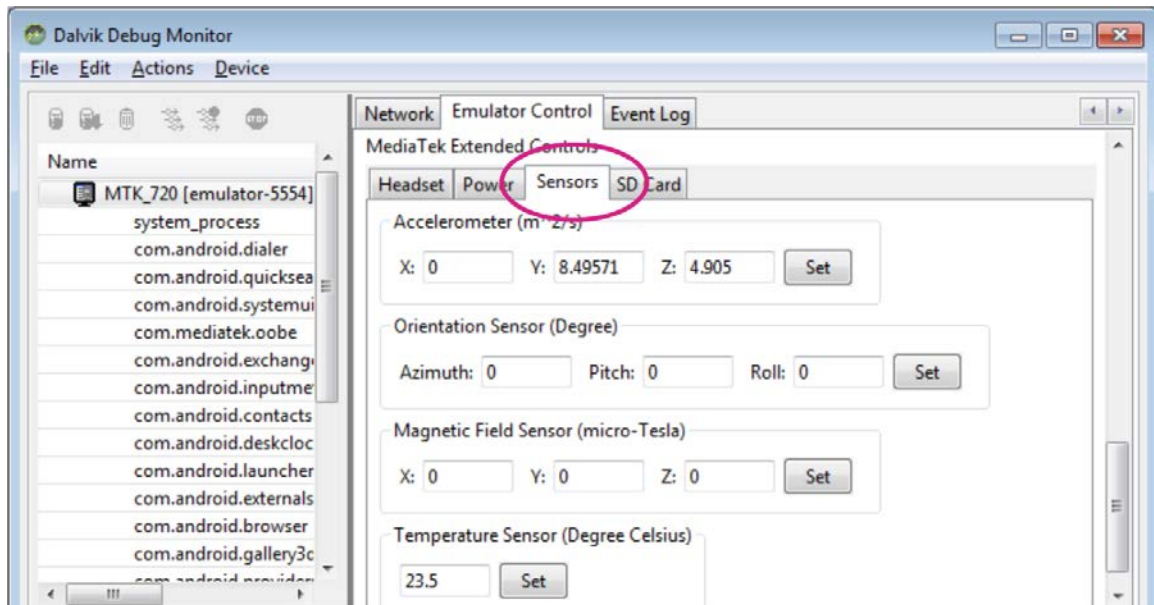


图 19 传感器的各种状态

4.8.7 模拟 SD 卡状态

如图 20 所示，在 **Emulator Controls** 中的 **SD Card** 选项下，点击 **Insert SD Card** 或 **Remove SD Card** 可以模拟是否有 SD 卡插入。

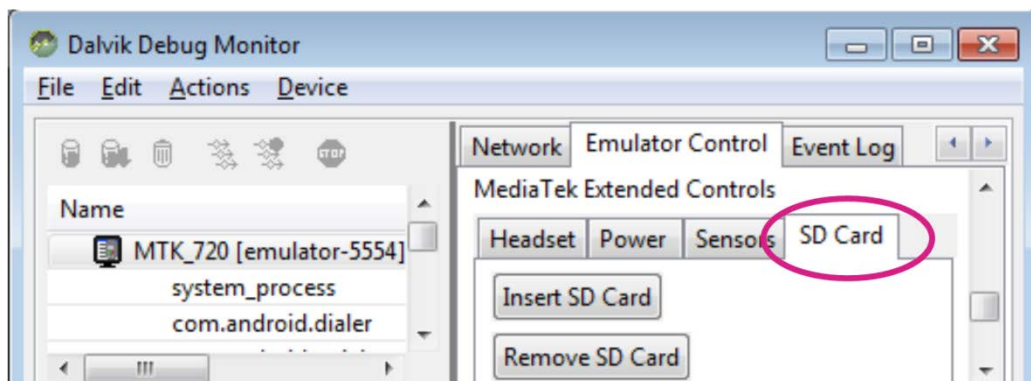


图 20 模拟插入或取出 SD 卡的选项

4.9 其他工具

本节描述当您使用联发科模拟器时其他可能有用的 Android SDK 工具。

4.9.1 Intel®硬件加速执行管理器

Intel®硬件加速执行管理器（HAXM）可以用来优化联发科模拟器的性能。你可以在 [Intel 网站](#) 上下载这个工具，并按照所提供的安装说明进行安装。

您可以按照以下检查是否已安装有 Intel®HAXM:

- 在 Windows 操作系统的命令提示符下输入 SC 查询 intelhaxm，如图 21 所示。

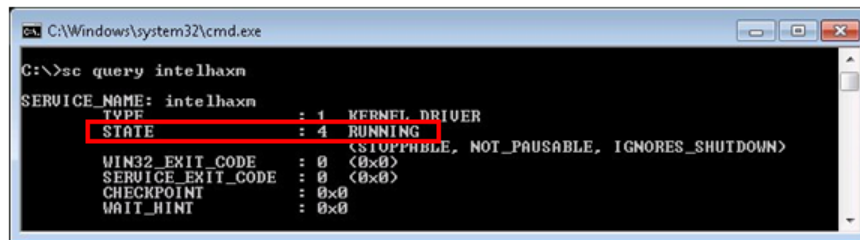


图 21 Windows 下查询 HAXM 是否存在

- 在 Linux 操作系统的终端窗口中输入 kvm-ok，如图 22 所示。

```

$ kvm-ok
INFO: Your CPU supports KVM extensions
INFO: /dev/kvm exists
    
```

图 22 Linux 下查询 HAXM 是否存在

如果无法获得上述结果则表示 IntelHAXM 尚未安装。

5 API 指南

本章提供了有关 API 级别的信息和使用以下三个关键 API 的指南：

- HotKnot (com.mediatek.hotknot)
- Multi-SIM (com.mediatek.telephony)
- MAV (com.mediatek.hardware)

5.1 API 级别和兼容性

像 Android SDK API 一样，联发科的 SDK API 是向后兼容的，所以您的应用程序应该始终工作在其开发的版本上或联发科 Android 平台的更新版本上。

每一个受支持的联发科设备都配备了联发科 SDK API 分别在级别 1，2，或 3. 如图 23 所示，采用级别 1 编写的程序能在联发科 SDK API 级别 1 或者更高级别的设备上运行。

尽管这是第一个针对开发者发布的 API，您仍会遇到支持级别 1 和级别 2 的设备，因为应用于其他设备供应商的早期 Android 版本已包含这些 API。

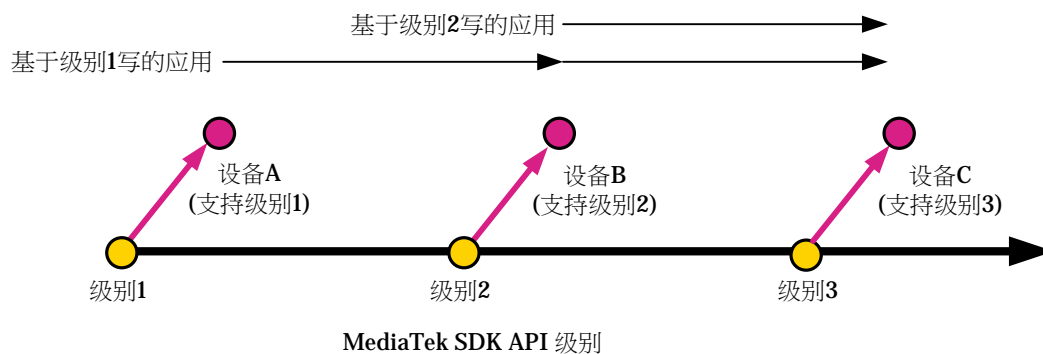


图 23 跨级别的 API 支持

在级别 1，2，和 3 中被支持的 API 如表表 3'所示

包	级别 1	级别 2	级别 3
com.mediatek.build	X	X	X
com.mediatek.telephony	X	X	X
com.mediatek.hardware	X	X	X
com.mediatek.media	X	X	X
com.mediatek.camcorder	X	X	X
com.mediatek.hotknot		X	X
com.mediatek.effect			X
com.mediatek.imagetransform			X
com.mediatek.gesture			X

表 3 按级别所示的联发科 API 支持

5.2 使用 HotKnot (com.mediatek.hotknot)

HotKnot 是使用电容式触摸屏功能使文件和数据得以设备之间传输的近场通信技术。操作范围一般是相距 0-1 厘米，1.43Kbps~7.88Kbps 的数据传输速率，提供半双工的串行传输。

5.2.1 创建 HotKnot 发送应用程序

使用 HotKnot 创建一个发送数据的应用程序需要：

- 1 创建一个包括 activities 的 Android 应用程序项目
- 2 在 AndroidManifest.xml 中添加：

```
<uses-permission android:name="android.permission.HOTKNOT"/>
```

- 3 初始化 HotKnot API 并检查设备是否启用 HotKnot

- a 获取 HotKnotAdapter

```
Adapter hotKnotAdapter = HotKnotAdapter.getDefaultAdapter(activity);
```

- b 检查 HotKnot 是否启用

```
if(HotKnotAdapter.isEnable()) {
    Log.d(TAG,"HotKnot is enable");
}
else {
    HotKnotAdapter.enable();
}
```

c 设置回调函数检测 HotKnot 是否传输完成

```
HotKnotAdapter.setOnHotKnotCompleteCallback {
    new HotKnotAdapter.OnHotKnotCompleteCallback() {
        public void onHotKnotComplete(int reason) {
            Log.d(TAG, "onHotKnotComplete reason:" + reason);
        }
    }, activity);
```

4 准备好要发送的消息：

o 案例 1：发送 URI

```
URI[] uri = {uri1, uri2};
HotKnotAdapter.setHotKnotBeamUri(uris, activity);
```

o 案例 2：发送短信

```
String mimeType = "binary/com.mediatek.demo";
byte[] payload = "Hello".getBytes();
HotKnotMessage message = new HotKnotMessage(mimeType, payload);
HotKnotAdapter.setHotKnotMessage(message, activity);
```

5 将 HotKnot 设备（发送器和接收器）的屏幕放在一起。

5.2.2 创建 HotKnot 接收应用程序

使用 HotKnot 创建一个接收数据的应用程序：

- 1 创建一个包括 activities 的 Android 应用程序项目。
- 2 在 AndroidManifest.xml 中添加：

```
<uses-permission android:name="android.permission.HOTKNOT"/>
```

3 添加 HotKnot 消息的过滤规则：

```
<activity
    android:name=".DemoActivity"
    android:label="@string/app_name">
    <intent-filter>
        <action android:name="com.mediatek.HotKnot.action.MESSAGE_DISCOVERED" />
        <data android:mimeType="binary/com.mediatek.demo" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

4 将 HotKnot 设备（发送器和接收器）的屏幕放在一起。

5 获取并解析 HotKnot 消息，直到目标活动已启动。

```
@Override
protected void onCreate(Bundle bundle) {
    super.onCreate(bundle);

    Intent intent = getIntent();
    If(HotKnotAdapter.ACTION_MESSAGE_DISCOVERED.equals(intent.getAction())) {

    }
    else {
        Log.w(TAG, "Unknown intent: " + intent);
        finish();
        return;
    }

    // 获取 HotKnot 消息
    String mimeType = intent.getType();
    HotKnotMessage HotKnotMessage =
intent.getByteArrayExtra(HotKnotAdapter.EXTRA_DATA);

    if("binary/com.mediatek.demo".equalsIgnoreCase(mimeType)) {
        String message = new String(HotKnotMessage);
        if("hello".equals(message)) {
            //正确
        }
    }
}
```

5.3 使用多 SIM (com.mediatek.telephony)

联发科平台提供了以下两种类型的多 SIM 功能：

SmsManagerEx 和 TelephonyManagerEx.

5.3.1 TelephonyManagerEx 类

TelephonyManagerEx 类提供了在设备上访问电话服务信息的功能，特别是配备有多 SIM 卡的设备。应用程序可以在这个类中使用该方法来确定电话的服务和状态，以及访问某些类型的用户信息。应用程序也可以注册一个监听器来接收电话的状态变化通知。

例如：如果应用程序想知道在特定的 SIM 通话状态，可以使用下面的代码：

```
TelephonyManagerEx.getDefault().getCallState(simID);
```



simID: 0 is for SIM1 phone call state; 1 is for SIM2 phone call state.

5.3.2 SmsManagerEx 类

SmsManagerEx 类是 SmsManager 的一个扩展。扩展的功能主要是使用了新加的变量 slotId 和 simId 来支持多 SIM 卡功能。如果你想使用多 SIM 卡功能的短信功能来开发应用程序，则可以使用 SDK 模拟器来测试这些功能。

下面是使用消息文本“Test words” 发送短信的代码：

```
String text = "Test words";/* 需要发送的文字*/
```

- 1 将句子分成多份，因为一个短信的大小是有限的。

```
// 将信息分成多个部分
messages = SmsManagerEx.getDefault().divideMessage(text, codingType);
int messageCount = messages.size();
```

- 2 要获得有关信息是否发送完成或交付完成的信息，需要在发送或传递消息的每一个部分创建 deliveryIntents 和 sentIntents，最终 intent 将被传出去。

```
/*
 * 在交付后，每一个部分都可以添加 deliveryIntent 和 sentIntent
 * 或发送完后，intent 将被发送出去。
 */

ArrayList<PendingIntent> deliveryIntents = new
ArrayList<PendingIntent>(messageCount);
ArrayList<PendingIntent> sentIntents = new ArrayList<PendingIntent>(messageCount);
for (int i = 0; i < messageCount; i++) {
    //一个短信只有最后一部分需要 deliveryReport
    if (mRequestDeliveryReport && (i == (messageCount - 1))) {
        Intent intent = new Intent("sms_test_delivery_intent");
        deliveryIntents.add(PendingIntent.getBroadcast(mContext, i, intent, 0));
    } else {
        deliveryIntents.add(null);
    }
    Intent intent = new Intent("sms_test_intent_sent_a_part");
    //通常我们可能需要知道发送信息的全部内容。
    if (i == messageCount - 1) {
        intent.putExtra("sms_test_key_is_last_part", true);
    }
    sentIntents.add(PendingIntent.getBroadcast(mContext, i, intent, 0));
    /// @}
}
```

- 3 将短信发送到目标地址时，需要将 scAddr 设置为 null，除非你需要指定一个服务中心。

```
//将多部分文本发送到目标
try {
    SmsManagerEx.getDefault().sendMultipartTextMessageWithEncodingType(
        destAddr/* targetAddress */, scAddr/* serviceCenter */, messages, codingType,
        sentIntents, deliveryIntents, slotId);
} catch (Exception ex) {
    throw new MmsException("SmsMessageSender.sendMessage: caught " + ex +
        " from SmsManager.sendMessage()");
}
```

5.4 使用 MAV 捕捉 (com.mediatek.hardware)

有一个类，它提供了联发科平台上的 MAV 功能：CameraEx。

5.4.1 CameraEx 类

CameraEx 类用于启动和停止多视角（MAV）的图像捕捉。您的应用程序在拍摄 MAV 照片时的工作流与普通照相的相类似，也就是调用 startMav 代替 takePicture，并在捕获的数量达到启用 MAV 捕捉时设置的图像数量后，调用 stopMAV。此外，您可以在拍摄过程中停止 MAV，然后将参数 isMerge 设置为 false 来放弃拍摄。

以下是记录 MAV 的代码：

1 实现 CameraEx.MavCallback 接口。

```
private final class MavFrameCallback implements MAVCallback {

    public void onFrame(byte[] jpegData) {
        // To Do
    }
}
```

2 添加一个用于数据回调的监听器。

```
mCamera.getCameraDevice().setMAVCallback(getMavCallback());
```

3 开始捕捉一系列图片放入 MAV 中。

```
mCamera.getCameraDevice().startMAV(NUM_MAV_CAPTURE);
```

NUM_MAV_CAPTURE: MAV 中图片的总数

4 停止 MAV 捕捉。

a 如果单独成像图片数量达到总集，则将 setmavcallback 设为 null。

```
If (isMerge) {
    mCamera.getCameraDevice().setMAVCallback(null);
}
```

b 否则，请在调用 stopMAV 之前确保使用相同的相机设备。

```
if (holder.isSameCameraDevice(mCamera.getCameraDevice(),mCamera.getCameraId())) {
    //表示 hw 已□被关□
    //并且不需要再用 stop 了。
    mCamera.getCameraDevice().stopMAV(isMerge ? 1:0);
} else {
    Log.w(TAG, "doStop device is release?");
}
```