Android 系统移植结构和方法

Android 系统移植结构和方法

- 第一部分 Android 的 Linux 操作系统
- 第二部分 Android 的硬件抽象层
- 第三部分 Android 中各个部件的移植方式

第一部分 Android 的 Linux 操作系统

- 1.1 标准的 Linux 操作系统
- 1.2 Android 对的 Linux 内核的使用
- 1.3 Linux 内核空间到用户空间的接口

1.1 标准的 Linux 操作系统

Android 是基于 Linux 操作系统的，Linux 操作系统包括 Linux 内核和驱动程序。Linux 操作系统位于 Android 软件系统的最低的一个层次。

Android 系统目前可以支持 ARM 平台的多种处理器，也可以支持 MIPS 和 x86 平台，这些平台之间的可移植性是由 Linux 操作系统的移植性来实现的。

1.1 标准的 Linux 操作系统

由于 Linux 可以支持众多的体系结构，因此 Linux 内核在众多的体系结构中的移植是一个重要的部分。Linux 将每种体系结构组成一个文件夹。

以 ARM 体系结构为例，又包含了各种不同的处理器，操作对于它们的代码又不相同。因此，在 arch/arm 下包含了两方面的代码：与处理器无关的公共代码以及与处理器相关的部分。

在 ARM 体系结构中，又分为不同的机器。每一种 ARM 体系对应 arch/arm 目录中一个名称为 mach-XXX 的文件夹。

1.1 标准的 Linux 操作系统

驱动程序也是 Linux 操作系统中的一个重要部分。在目前的 Linux 内核的源代码中，驱动程序占据了大部分。在 Linux 操作系统中，系统调用是应用程序和内核（kernel）之间的接口，而设备驱动程序是操作系统内核和机器硬件之间的接口。设备驱动程序为应用程序屏蔽了硬件的细节，这样在应用程序看来，硬件设备通常是一个标准的设备文件，应用程序可以像操作普通文件一样对硬件设备进行操作。

1.2 Android 对的 Linux 内核的使用

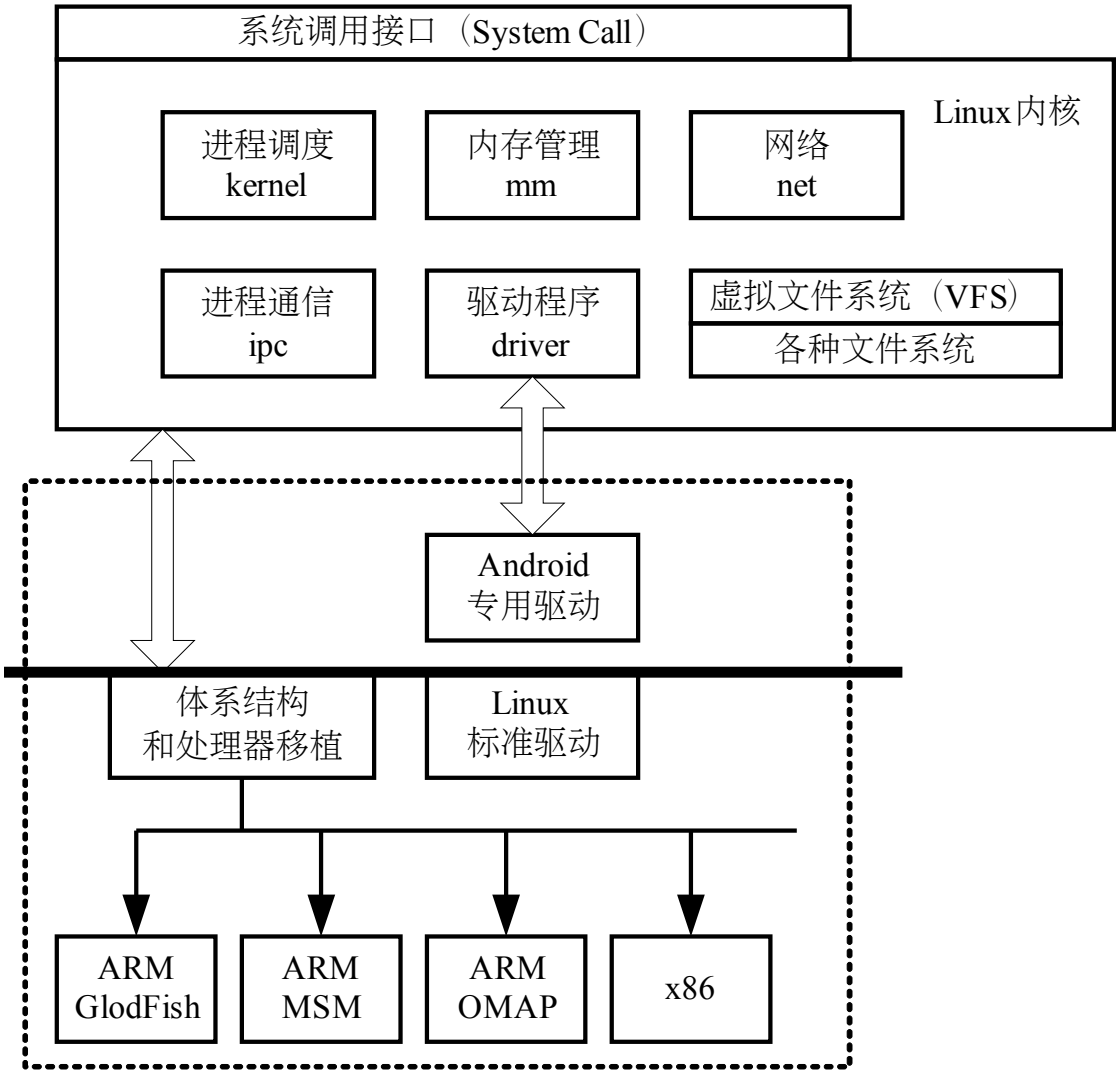
Android 中使用 Linux 操作系统，除了 Linux 的通用代码之外，主要包含 3 个方面的内容：

1 体系结构和处理器相关

2 Android 特定的驱动程序

3 标准的设备驱动程序

1.2 Android 对的 Linux 内核的使用



1.2 Android 对的 Linux 内核的使用

Android 中 Linux 操作系统的 3 个方面中：体系结构处理器和标准的设备驱动程序这两个方面是和硬件相关，但是对于同一种硬件，在 Android 系统和非 Android 的 Linux 系统中是基本一样的； Android 特定的驱动程序，通常是和硬件无关的驱动程序，仅仅在 Android 系统中使用，但是同于同样适用 Android 操作系统的不同硬件，这部分的内容是一样的。

Android 系统通常用于移动设备或者其他的嵌入式设备，因此多基于 ARM 体系结构，在 ARM 体系结构具有多种处理器。对于同一种处理器，对于不同外围设备，因此可能也将使用不同的驱动程序。

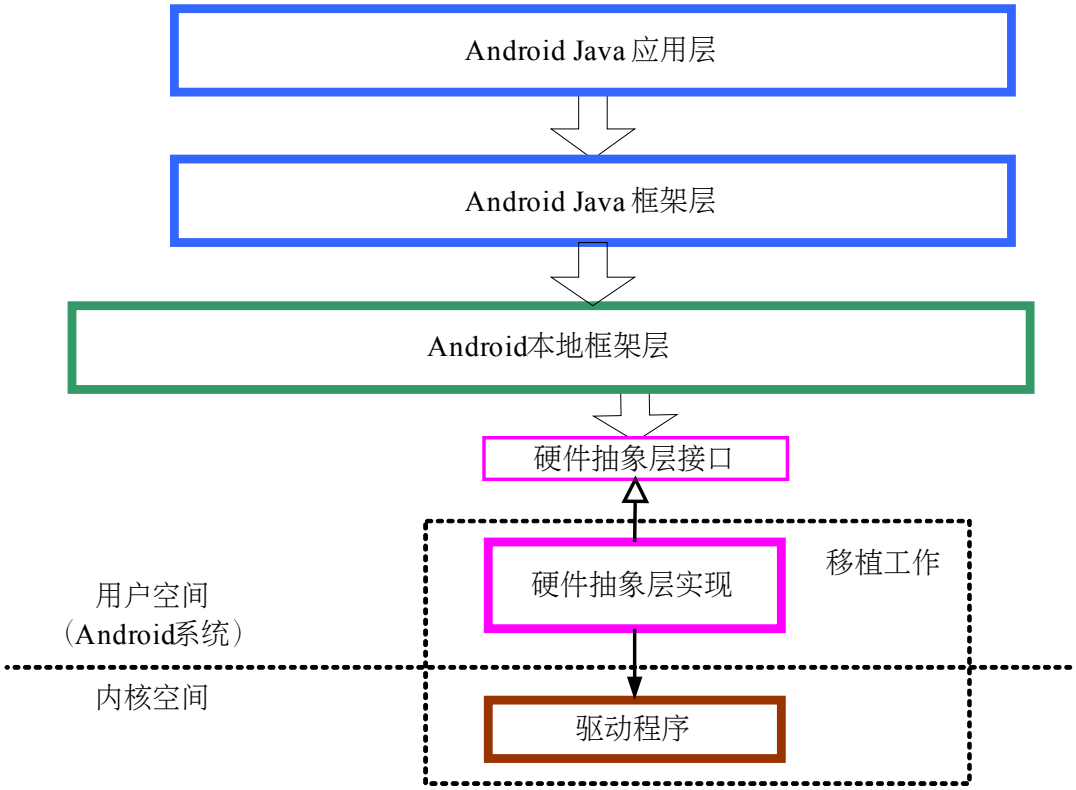
1.2 Android 对的 Linux 内核的使用

为 Android 构建一个基本 Linux 操作系统，如果以非 Android 的 Linux 操作系统为起点，那么主要的工作就是增加 Android 特定的驱动程序。 Android 中的 Linux 操作系统包含了很多的驱动程序，将其移植到一个新的系统中的步骤比较简单：

- 增加源代码
- 在 KConfig 中增加内容
- 在 Makefile 中增加内容

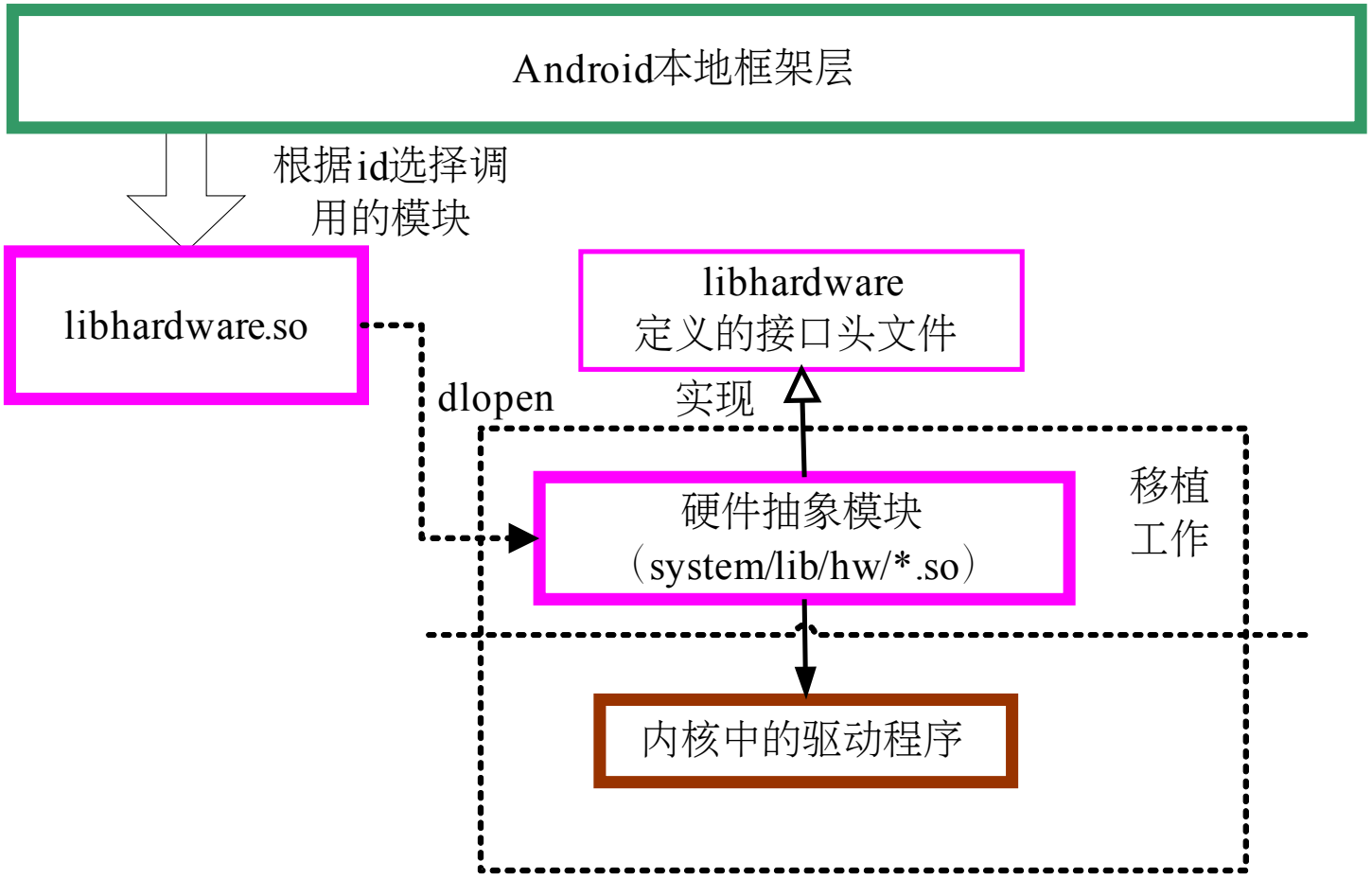
第二部分 Android 的硬件抽象层

硬件抽象层是位于用户空间的 Android 系统和位于内核空间的 Linux 驱动程序中间的一个层次。

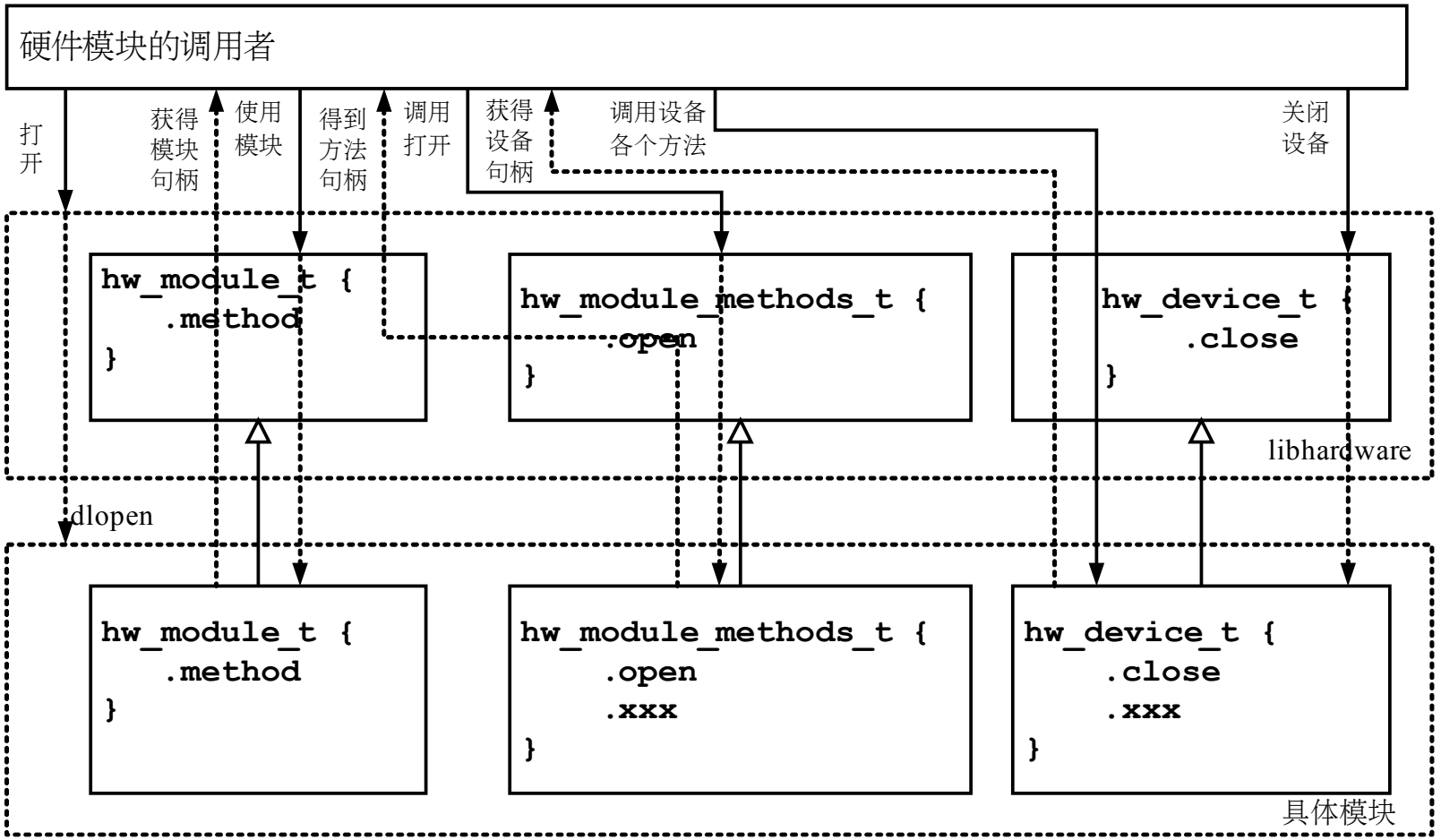


第二部分 Android 的硬件抽象层

hardware 模块的方式



第二部分 Android 的硬件抽象层



第二部分 Android 的硬件抽象层

直接接口方式

hardware_legacy 库中的提供了一些各自独立的接口，由用户实现后，形成库，被直接连接到系统中。这是实现硬件抽象层最简单也是最直接的方式。 hardware_legacy 的头文件路径为：

hardware/libhardware_legacy/include/hardware_legacy

Android 中的蓝牙库 bluedroid 与之类似，也是采用同样的方式，其头文件的路径为：

system/bluetooth/bluedroid/include/bluedroid/bluetooth.h

hardware_legacy 库中包含了几个 C 接口的文件，如 GPS ，power， wifi， vibrator 等，同时在 hardware_legacy 库中也包含了 power， wifi， vibrator 这几个系统的实现和 GPS 在仿真器上的实现。在开发一个新的硬件系统的时候，可以根据需要去实现这几个库，也可以使用系统默认的实现方式。

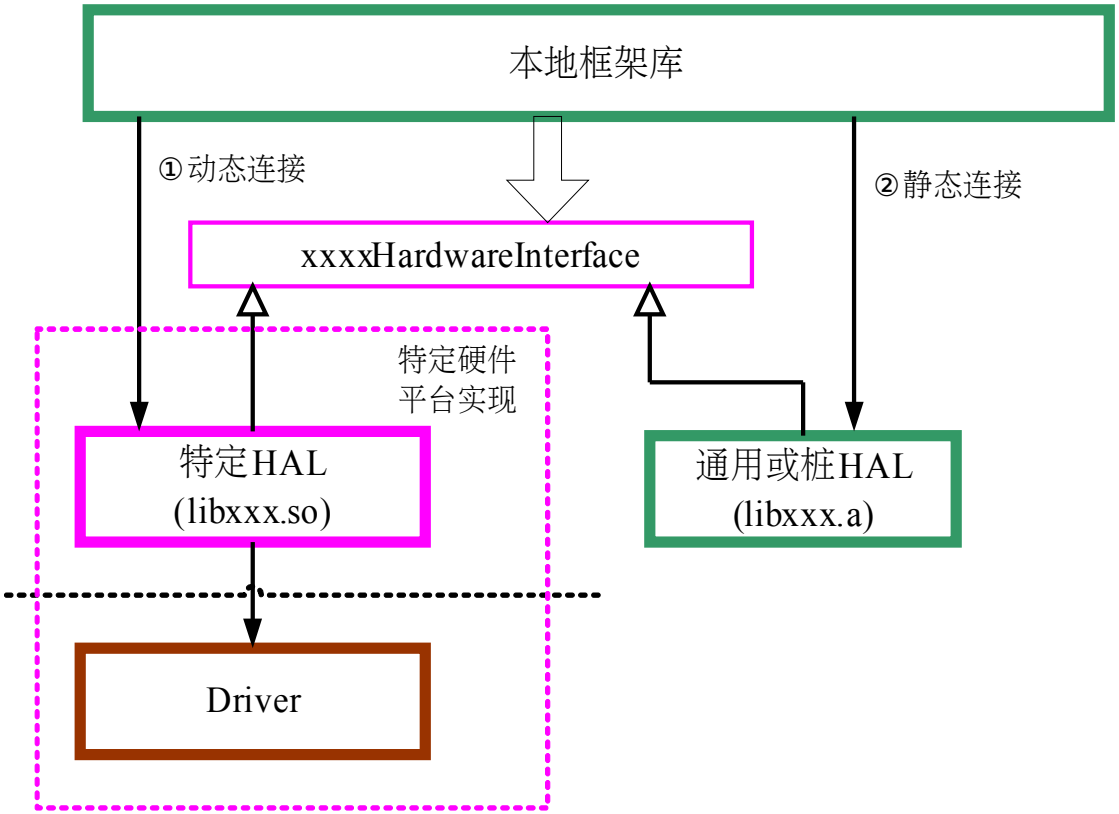
第二部分 Android 的硬件抽象层

C++ 的继承实现方式

使用 C++ 的类的继承方式实现硬件抽象层，也是 Android 中的一种方式。为了使用这种方式， Android 平台定义了 C++ 的接口。由具体的实现者继承实现这些接口。

在 Android 系统中，通常也有通用的实现方式，可以作为一个简易的实现或者“桩（ stub ）”的作用。

第二部分 Android 的硬件抽象层



第二部分 Android 的硬件抽象层

直接调用驱动

在 Android 中有一些比较简单的子系统，并没有在“物理”上存在的硬件抽象层，也就是说，实现其硬件抽象功能的部分不在单独的代码中。例如：由 JNI 部分代码直接调用的驱动程序的设备节点或者使用 sys 文件系统。

Alarm 子系统就是采用的这种方式，在 JNI 中直接调用的驱动程序。此外，在 Android 的 JNI 代码中，还包含一些对 sys 文件系统的访问，例如 switch 等。

第三部分 各个部件的移植方式

显示（旧） Framebuffer 标准 DisplaySurface （Android 标准）
显示（新） Framebuffer 标准或其他 Gralloc 硬件模块
用户输入 Event 设备 EventHub （Android 标准）
3D 加速（旧） 非标准 OpenGL 抽象层（Android 标准）
3D 加速（新） 非标准 OpenGL 抽象层（Android 标准）
音频 非标准（Alsa 、OSS 或其他） C++ 继承的硬件抽象层
视频输出 非标准（fb， v4l2 或其他） overlay 硬件模块
摄像头 非标准（多为 v4l2） C++ 继承的硬件抽象层
多媒体编解码 非标准 Skia 和 OpenMax 插件
电话 非标准 动态开发的插件库

第三部分 各个部件的移植方式

全球定位系统 非标准（多为 UART ） 直接接口
无线局域网 标准 Wlan 驱动
Wpa （ Linux 标准） 和 wifi 库 （ Android 标准）
蓝牙 标准 Bluetooth 驱动
Bluez （ Linux 标准和 Bluedroid 库 （ Android 标准）
传感器 非标准 Sensor 硬件模块
位块复制 非标准 Copybit 硬件模块
振动器 Sys 文件系统的固定位置 直接接口 （ Android 标准）
背光和指示灯 非标准 Light 硬件模块
警告器 Misc 设备 （ Android 标准） 简化到 JNI 层次中
电池管理 Sys 文件系统的固定位置 直接使用接口 （ Android 标准）

谢谢！