

深圳易新泰科技有限公司 easitek



深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

H618 Android 开发指南

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

版本号: 1.0
发布日期: 2022.05.07

深圳易新泰科技有限公司 easitek

深圳易新泰科技有限公司 easitek

深圳易新泰科技

版本历史

版本号	日期	制/修订人	内容描述
1.0	2022.05.07	AWA1829	初始版本文档



目 录

1 前言	1
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
2 Android 开发环境	2
2.1 开发环境	2
2.2 代码下载	2
3 规格说明	3
4 基础组件说明	4
4.1 Activity 组件	4
4.2 Service 组件	4
4.3 BroadcastReceiver 组件	4
4.4 ContentProvider 组件	4
4.5 Context	5
4.6 Application	5
4.7 Intent	5
5 快速开发配置指南	6
5.1 方案文件替换	6
5.2 overlay 说明	6
5.2.1 为产品添加 Overlay 目录	6
5.2.2 改变 mk 文件来添加 overlays 的编译项	7
5.2.3 在 overlay 目录下创建资源文件	7

1 前言

1.1 文档简介

本文档介绍 Android 12 系统开发环境、规格说明、基础组件、快速开发配置指南等内容。

1.2 目标读者

Android 开发人员。

1.3 适用范围

Allwinner H618 Android 12 平台。

2 Android 开发环境

2.1 开发环境

1. 系统环境：Ubuntu 20.04（非指定版本）
2. 安装 JDK（官网下载地址：<https://www.oracle.com/java/technologies/downloads/>）

2.2 代码下载

按照以下步骤下载 android 代码。

```
mkdir android
cd android
repo init -u 清单库地址
repo sync
```

3 规格说明

目前，H618 主要的两款板型为 P1 和 P2 板，对应的规格如下表。

芯片代号	board	longan kernel	longan arch	android 配置	内存	wifi
H618	P1	linux-5.4	arm64	apollo_p1	4GB	AW869A
H618	P2	linux-5.4	arm64	apollo_p2	2GB	AW869A

4 基础组件说明

4.1 Activity 组件

Activity 是 Android 最重要的组件，负责用户界面的设计。Activity 用户界面框架采用 MVC 模式。

- 控制器：负责接受并响应程序的外部动作。
- 通过视图：反馈应用程序给用户的信息 (UI 界面反馈)。
- 模型：应用程序的核心，用于保存数据和代码。

Activity 之间通过 Intent 进行通信。

4.2 Service 组件

Service 是一个应用程序的组件，长时间运行在后台，不提供用户的交互界面。当应用程序退到后台，运行的服务不会受到影响，继续在后台中运行。

一个应用程序可以启动另一个应用的服务或者直接绑定另一个应用的服务直接实现进程间的通信，可以用来开发如监控类程序。

4.3 BroadcastReceiver 组件

在 Android 系统中，BroadcastReceiver 组件实现监听/接收应用 App 发出的广播消息。

- 例如，开机启动完成、收到短信、电池电量改变、网络状态改变等。

BroadcastReceiver(广播接收者) 接收来自系统或其他应用程序的广播，并做出回应。

4.4 ContentProvider 组件

为了跨进程共享数据，Android 提供了 ContentProvider 接口，可以在无须了解数据源、路径的情况下，对共享数据进行查询、添加、删除和更新等操作。

4.5 Context

在应用程序中 Context 的具体实现子类就是：Activity，Service，Application

- Context 数量 = Activity 数量 + Service 数量 + 1

Context 提供了关于应用环境全局信息的接口。它是一个抽象类，它的执行被 Android 系统所提供。

4.6 Application

当 Android 程序启动时系统会创建一个 Application 类型的对象，用来存储系统的一些信息完成数据传递、共享和缓存等操作。

Application 对象的生命周期是整个程序中最长的，它的使用寿命就等于这个程序的使用寿命，且是全局、单例的，即在不同的 Activity 和 Service 中获得的对象都是同一个对象。

4.7 Intent

Android 提供轻量级的进程间通信机制 Intent，使跨进程组件通信和发送系统广播成为可能，组件 Activity、Service 和 BroadcastReceiver 都是通过消息机制被启动，其使用的消息就封装在对象 Intent 里。

5 快速开发配置指南

可以通过 `cdevice` 命令或者 `cd $android_device_path` 进入 Android 方案配置目录 (`device/{vendor-name}/{platform-name}`)

各项配置文件路径请参考 `device/{vendor-name}/{platform-name}/README.md` 说明。

5.1 方案文件替换

Android 方案中有时需要定制化一些需求，如替换开机 LOGO、`env.cfg`、分区配置等，除了可在 `longan` 目录下对应的路径下替换外，android 方案也提供了接口进行替换。其原理是在 `pack` 操作时，将需要替换的文件拷贝到 `longan/out/pack_out` 目录下。

在 android 方案中通过 `BOARD_ADD_PACK_CONFIG` 配置来指定需要替换的文件：

```
# BoardConfig.mk
BOARD_ADD_PACK_CONFIG += $(TARGET_DEVICE_DIR)/system/sys_partition.fex
BOARD_ADD_PACK_CONFIG += $(PRODUCT_PLATFORM_PATH)/common/system/bl_env.cfg:env.cfg
```

通过这种方式替换的文件只能是会被拷贝到 `longan/out/pack_out` 目录下的文件。格式为：`local_filepath[:copy_filename]`。

5.2 overlay 说明

Android overlay 机制允许在不修改 `apk` 或者 `framework` 源代码的情况下，实现资源的定制。以下几类能够通过 overlay 机制定义：

1. Configurations (string, bool, bool-array)
2. Localization (string, string-array)
3. UI Appearance (color, drawable, layout, style, theme, animation)
4. Raw resources (audio, video, xml)

更详细的资源文件可浏览[android 网站](#)

5.2.1 为产品添加 Overlay 目录

有两种不同的 overlay 目录定义：

1. PRODUCT_PACKAGE_OVERLAYS

用于指定产品

2. DEVICE_PACKAGE_OVERLAYS

用于同一设备模型的一系列产品

如果包含同一资源，那么 PRODUCT_PACKAGE_OVERLAYS 的内容将覆盖 DEVICE_PACKAGE_OVERLAY 中的内容。如果要定义多个 overlays 目录，需要用空格隔开，同一资源的定义，将使用先定义的目录中的资源。

在方案目录下创建 common/overlay 和 {device-name}/overlay 目录，分别用于 device 通用及 product 使用的 overlay 文件夹。

5.2.2 改变 mk 文件来添加 overlays 的编译项

在文件 common/overlay/config.mk 和 {device-name}/overlay/config.mk 中分别添加：

```
DEVICE_PACKAGE_OVERLAYS := \  
$(LOCAL_MODULE_PATH)/overlay \  
$(DEVICE_PACKAGE_OVERLAYS)
```

```
PRODUCT_PACKAGE_OVERLAYS := \  
$(LOCAL_MODULE_PATH)/overlay \  
$(PRODUCT_PACKAGE_OVERLAYS)
```

注：必须加上 \$(PRODUCT_PACKAGE_OVERLAYS) 变量否则将找不到默认资源。

5.2.3 在 overlay 目录下创建资源文件

在 overlay 目录下创建和要替换资源所在文件相同的路径的文件，此路径是相对于 android platform 目录。如替换 framework-res 路径为：platform/framework/base/core/res/res/values/config.xml 中的某一项，则在 overlay 中创建对应的路径：overlay/framework/base/core/res/res/values/config.xml 并添加要修改的一向配置，如：

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <bool name="config_showNavigationBar">true</bool>  
</resources>
```

著作权声明

版权所有 © 2022 珠海全志科技股份有限公司。保留一切权利。

本档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本档内容的部分或全部，且不得以任何形式传播。

商标声明

、 **全志科技** （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本档作为使用指导仅供参考。由于产品版本升级或其他原因，本档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本档中提供准确的信息，但并不确保内容完全没有错误，因使用本档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。