

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

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

# H618 Android 12 SDK 快速开发指南

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

版本号: 1.0  
发布日期: 2022.05.07

## 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

<b>1 前言</b>	<b>1</b>
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
<b>2 Android 12 SDK 目录结构</b>	<b>2</b>
<b>3 编译指南</b>	<b>4</b>
3.1 编译 32 位 Android 固件	4
3.2 编译 64 位 Android	5
3.3 编译 Linux/dragonboard	6
3.4 编译 brandy (默认编译 longan 已经编译 brandy)	6
3.5 注意事项	7
3.5.1 版权声明	7
3.5.2 内核编译输出路径	7
3.5.3 更改内核配置	8
3.5.4 编译输出的固件路径	8
<b>4 烧写指南</b>	<b>9</b>

# 1 前言

## 1.1 文档简介

基于 Android 12 的 SDK 介绍 SDK 快速开发指南，适用于所有 H618 Android 12 平台。

## 1.2 目标读者

SDK 平台负责人，版本集成人员、SDK 开发人员、模块开发人员。

## 1.3 适用范围

Allwinner H618 Android 12。

## 2 Android 12 SDK 目录结构



test	
toolchain	
tools	
vendor	
aw	全志提供的服务与应用程序
partner_gms	GMS包 (仅向MADA客户单独开放)
partner_modules	Mainline模块 (仅向MADA客户单独开放)
security	安全密钥存放目录

### ⚠ 警告

为方便代码维护, 减少不必要的编译过程, Android11 之后, 将 Longan 相关代码放到了 android 目录下。除目录结构发生变化外, 其余使用方法保持一致。

### 📖 说明

Android12 开始, device/softwinner 目录下, 一个芯片仅维护一个仓库。如需添加产品方案, 在仓库中新建产品对应的 makefile 和目录即可。



## 3 编译指南

### 3.1 编译 32 位 Android 固件

```
$ cd <android_top_path>/longan
$ ./build.sh config      # ----longan配置

Welcome to mkscrip setup progress
All available platform:
  0. android
  1. linux
Choice [android]: 0
All available ic:
  0. h618
Choice [h618]:
All available board:
  0. ft
  1. p1          # ----H618 4GB内存方案, AW869A方案
  2. p2          # ----H618 2GB内存方案, AW869A方案
  4. perf1
  5. perf2
  6. perf3
  7. qa
Choice [p1] 1
All available flash:
  0. default
  1. nor
Choice [default]: 0
All available kern_ver:
  0. linux-5.4
Choice [linux-5.4]: 0
All available arch:
  0. arm          # ----32位内核, 根据实际需求选用, go系统必须使用32位内核
  1. arm64        # ----64位内核, 根据实际需求选用, 非Go系统使用64位内核
Choice [arm64]: 1
$ ./build.sh      # ----编译longan
$ cd <android_top_path>
$ source build/envsetup.sh # ----导入环境变量
$ lunch <PRODUCT>-<VARIANT> # ---选择对应的方案
$ make -j16       # ---编译Android, j开启多核编译, 服务器开发一般为服务器cpu数量的一
半
$ pack <-d> <-v> # ----打包固件
```

#### 📖 说明

**pack** 参数:

- d # --卡打印
- v # --安全固件

 说明

**Android12** 中增加了内核位数选项，因为 **Android12 Go** 要求必须使用 **32** 位内核。**Android11** 及以前 **A133** 仅使用 **64** 位内核

请先在 **longan** 目录执行 **./build.sh config**，然后在 **Android** 根目录执行 **source && lunch** 操作。**Android lunch** 依赖内核位数选择结果

 说明

**Android** 安全固件使用预生成的 **key**，**key** 默认放置在 **<android-top-path>/vendor/security/toc\_keys**，无需手动生成也可以打包安全固件。

 说明

**extract-bsp** 命令已经弃用，无需在编译完 **longan** 后执行，目前仅仅留空这个命令，后续会删除。

## 3.2 编译 64 位 Android

### 1. 修改目标架构 TARGET\_ARCH 为 64 位

```
$ cd android/device/softwinner/ceres
$ android/device/softwinner/ceres$ git diff
diff --git a/common/system/config.mk b/common/system/config.mk
index 13266ff..ae91bad 100644
--- a/common/system/config.mk
+++ b/common/system/config.mk
@@ -86,7 +86,7 @@ PRODUCT_PACKAGES += \
 # call other makefile
 # 32bit android,you should define TARGET_ARCH := arm
 # 64bit android,you should define TARGET_ARCH := arm64
-TARGET_ARCH ?= arm
+TARGET_ARCH ?= arm64
 ifeq ($(TARGET_ARCH),arm)
 $(call inherit-product, $(LOCAL_MODULE_PATH)/ceres_32_bit.mk)
 else ifeq ($(TARGET_ARCH),arm64)
```

### 2. 重新 source 以及选择配置方案，64 位 Android 建议在 2GB 内存以上的方案运行

```
$ source build/envsetup.sh && lunch
```

### 3. 如何确认已经配置为 64 位？lunch 后的输出如下配置即为 64 位 Android

```
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=generic
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv8-a
TARGET_2ND_CPU_VARIANT=generic
```

### 4. 其他编译配置方法与 32 位一致

 说明

**H618 Android12** 版本暂不支持 **64** 位 **Android**

### 3.3 编译 Linux/dragonboard

```
$ cd <android-top-path>/longan
$ ./build.sh config

Welcome to mkscrip setup progress
All available platform:
  0. android
  1. linux
Choice [android]: 1
All available linux_dev:
  0. bsp
  1. dragonboard
Choice [bsp]: 0
All available ic:
  0. a133
  1. h618
Choice [h618]: 1
All available board:
  0. ft
  1. p1
  2. p2
  3. perf1
  4. perf2
  5. perf3
  6. qa
Choice [p1]: 1
All available flash:
  0. default
  1. nor
Choice [default]: 0

$ ./build.sh # 编译Linux/dragonboard
$ ./build.sh pack<_debug><_secure> # 打包固件
```

#### 📖 说明

##### pack 参数:

- `_debug` # --卡打印
- `_secure` # --安全固件

#### 📖 说明

如果是第一次打包安全固件，需要手动在 `<android-top-path>/longan/build` 目录下运行 `./createkeys`，生成 `key`，然后再执行打包命令

### 3.4 编译 brandy（默认编译 longan 已经编译 brandy）

```
$ cd <android-top-path>/longan
$ ./build.sh config # 编译bootloader需先完成配置工作
$ ./build.sh bootloader # 编译uboot的命令
```

## 3.5 注意事项

### 3.5.1 版权声明

第一次使用 sdk 编译 Android 执行 source 命令，或编译 longan 执行 ./build.sh 相关命令时，均会有如下提示，请务必仔细阅读提示并选 Y 接受：

```
Before using this files, please make sure that you note the following important information
.
*****

Copyright (c) 2019-2022 Allwinner Technology Co., Ltd. ALL rights reserved.

Allwinner is a trademark of Allwinner Technology Co.,Ltd., registered in
the the People's Republic of China and other countries.
All-Allwinner Technology Co.,Ltd. trademarks are used with permission.

DISCLAIMER
THIRD PARTY LICENCES MAY BE REQUIRED TO IMPLEMENT THE SOLUTION/PRODUCT.
IF YOU NEED TO INTEGRATE THIRD PARTY'S TECHNOLOGY (SONY, DTS, DOLBY, AVS OR MPEGLA, ETC.)
IN ALLWINNERS' SDK OR PRODUCTS, YOU SHALL BE SOLELY RESPONSIBLE TO OBTAIN
ALL APPROPRIATELY REQUIRED THIRD PARTY LICENCES.
ALLWINNER SHALL HAVE NO WARRANTY, INDEMNITY OR OTHER OBLIGATIONS WITH RESPECT TO MATTERS
COVERED UNDER ANY REQUIRED THIRD PARTY LICENSE.
YOU ARE SOLELY RESPONSIBLE FOR YOUR USAGE OF THIRD PARTY'S TECHNOLOGY.

THIS SOFTWARE IS PROVIDED BY ALLWINNER"AS IS" AND TO THE MAXIMUM EXTENT
PERMITTED BY LAW, ALLWINNER EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND,
WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING WITHOUT LIMITATION REGARDING
THE TITLE, NON-INFRINGEMENT, ACCURACY, CONDITION, COMPLETENESS, PERFORMANCE
OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
IN NO EVENT SHALL ALLWINNER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS, OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.

*****

You can read <xxxx>/disclaimer/Allwinnertech_Disclaimer(Cn_En)_20181122.md for detailed
information.

You read time left 8 seconds....
I have already read, understood and accepted the above terms? [Y/N]
```

### 3.5.2 内核编译输出路径

为维护干净整洁的代码环境，Linux-5.4 开始，将内核编译的输出结果保存到了 <android-top-path>/longan/out/kernel/build 下面。所有的.o、驱动 ko、vmlinux、.config 等文件请到该目录下找。

### 3.5.3 更改内核配置

如需将 defconfig 文件重新载入到.config，请执行如下命令，而不是直接将默认 defconfig 文件拷贝到.config：

```
$ ./build.sh loadconfig
```

如需更改内核.config 配置，请执行如下命令代替原来的 make menuconfig 命令：

```
$ ./build.sh menuconfig
```

更改完成，如需保存到原有 defconfig 文件，请执行如下命令，而不是直接将.config 拷贝到默认 defconfig 文件：

```
$ ./build.sh saveconfig
```

### 3.5.4 编译输出的固件路径

pack 完成后，固件将会保存在 <android-top-path>/longan/out/下面。

## 4 烧写指南

安装 PhoenixSuit，使用指南请查看《PhoenixSuit 使用指南》。



## 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。