



H616 Android Q[®]

SDK Quick Start Guide

1.0
2019.9.21

文档履历

版本号	日期	制/修订人	内容描述
1.0	2019.9.21		

目录

1. 前言	1
1.1 编写目的	1
1.2 适用范围	1
1.3 相关人员	1
1.4 相关术语	1
2. SDK 概述	2
2.1 硬件资源	2
2.2 软件资源	4
2.2.1 安装 JDK	4
2.2.2 安装平台支持软件	4
2.2.3 安装 phoenixSuit	5
2.2.4 其他软件	5
2.3 代码下载说明	5
2.4 代码与 AOSP 差异比较	5
2.5 代码编译流程	6
2.5.1 内核编译流程	6
2.5.2 Android 代码编译流程	8
2.5.3 jack 编译问题	8
2.5.4 刷机方法	9
3. Declaration	10

1. 前言

1.1 编写目的

本文档介绍 H616 AndroidQ 系统方案快速上手编译刷机，以帮助客户快速熟悉开发环境，加快产品上市。

1.2 适用范围

系统定制化开发，系统调试

1.3 相关人员

Android 系统工程师、应用工程师

1.4 相关术语

- **homlet**: 面向客厅设备的产品线的名称。
- **SDK**: 全志定制环境总称
- **cupid**: 基于全志 H616 芯片的家庭娱乐产品代号
- **AOSP**: Android 系统开源代码项目

2. SDK 概述

2.1 硬件资源

- cupid 样机验证板一块
- 串口线一根 (TTL 电平)
- 双头 USB Type-A 连接线一根
- HDMI 连接线一根
- 5V 电源适配器一个

硬件上电连接方式如下, 如果开发板有 UART 接口可以直接连接 UART 接口, 如果没有 UART 接口可使用 TF 卡口复用, 可以用转接板配合烧写 card0 的固件使用 uart 调试, 开发主机建议使用 I7/E5 以上配置, 硬盘容量 >500G、内存容量 >8G

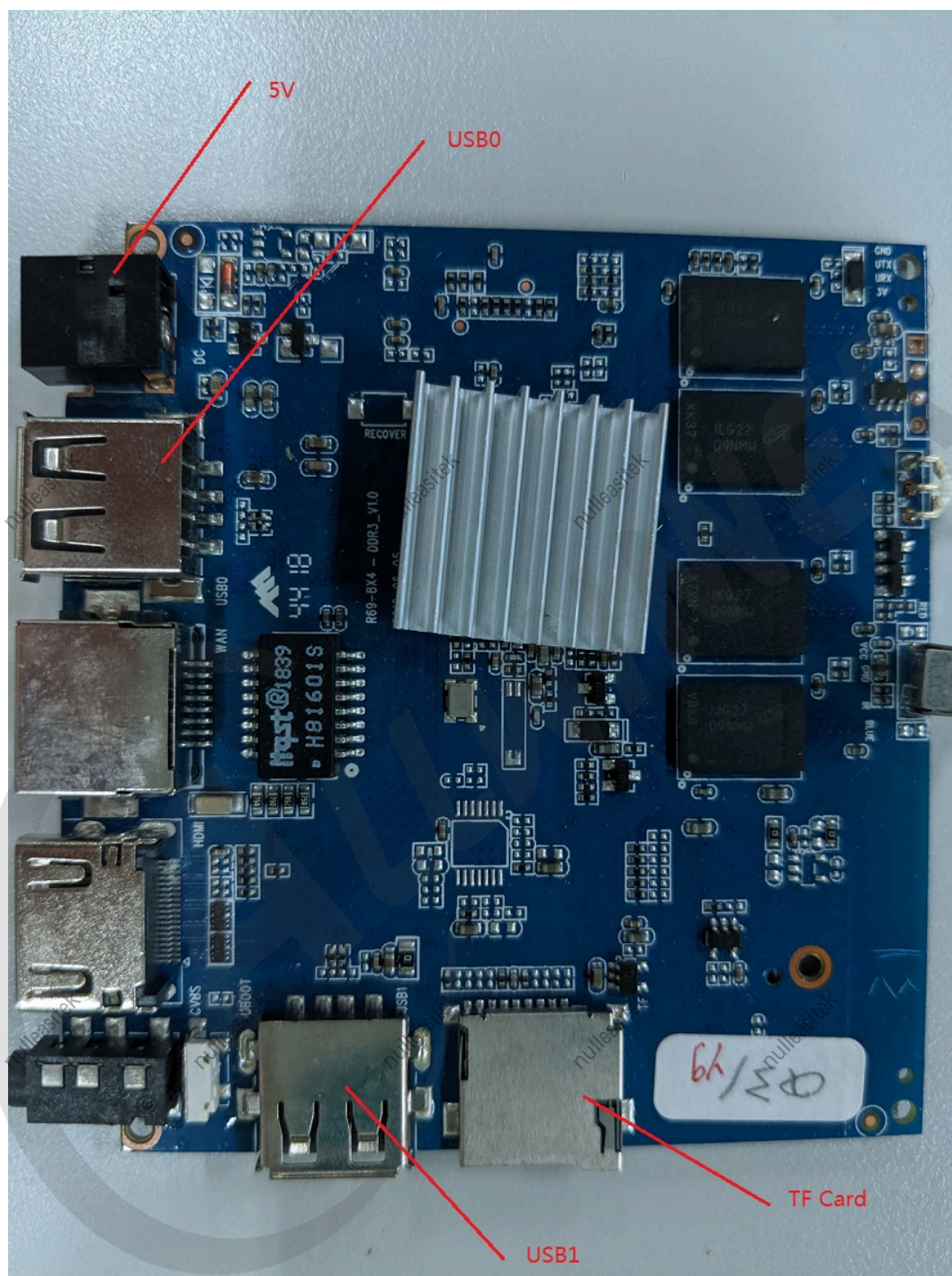


图 1

2.2 软件资源

由于 AndroidQ 编译环境已经不再支持 Ubuntu14.04 以下的版本，请务必使用 Ubuntu14.04 及以上的版本。避免不必要的编译错误

2.2.1 安装 JDK

AndroidN 以上版本的开发只能使用 openjdk8 的版本，高于或低于此版本以及 oracle 的 JDK 都会导致编译错误失败 openjdk-8 的安装方法有两种，一是通过命令

```
sudo apt-get install openjdk-8-jdk
```

二是下载已经打包好的 java-8-openjdk-amd64 archives 然后自行设置环境变量。这种方法可以实现多套 JDK 共存对于第一种方法，需要注意如果已经安装了其他的 JDK 版本，需要彻底删除，通过命令切换版本的方法并不能完全保证工具一致，推荐使用第二种方法安装，建议安装到如下路径：

```
/usr/lib/jvm/java-8-openjdk-amd64
```

在此路径下编译前，编译脚本会自动设置编译环境的 JDK 为 OpenJDK8

2.2.2 安装平台支持软件

Ubuntu 14.04 及以上版本使用如下命令安装

```
sudo apt-get install git-core gnupg flex bison gperf build-essential \
zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 \
lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z-dev ccache \
libgl1-mesa-dev libxml2-utils xsltproc unzip
```

安装 uboot 工具软件

```
sudo apt-get install u-boot-tools
```

2.2.3 安装 phoenixSuit

phoenixSuit 软件可以在 SDK 代码以下路径找到
longan/tools/tools_win/USB_update_and_produce/, 将 PhoenixSuit.msi 复制到 XP 主机上, 按照安装向导提示安装, 此软件支持 Windows XP SP3 及以上系统, 在 Ubuntu 环境下可以通过 Virtual Box 虚拟机使用之

2.2.4 其他软件

建议安装 Winrar 便于制作开机动画, 在 Windows 环境下可以安装 Windows Media Player 以使用 MTP 的功能

2.3 代码下载说明

请参考 SDK 发布文档的下载说明, 须向全志申请下载 sdk 的权限和账号。

2.4 代码与 AOSP 差异比较

SDK android 部分的代码是在 AOSP 的基础上开发的, 和 AOSP 的代码相比, 增加了以下的仓库

```
android/device/softwinner/common #系统公共代码, 框架层扩展的API
android/device/softwinner/cupid-common #H616公共配置文件
android/device/softwinner/cupid-p2 #H616 P2方案板配置仓库
android/frameworks/opt/net/pppoe #PPPOE支持代码
android/hardware/aw #全志平台HAL层代码仓库
android/hardware/realtek
android/hardware/xradio
android/vendor/aw/homlet #全志应用仓库
```


与 android 平级的 longan 是内核、uboot、硬件配置的目录，其主要内容如下：

```
longan/brandy #uboot boot0代码目录
longan/build #内核编译工具链及脚本目录
longan/kernel/linux-4.9 #内核代码
longan/device #方案硬件配置，打包工具等
```

对于客户自己建立的方案配置，建议放在 android/device/softwinner 目录下，硬件配置放在 longan/device/config/chips/h616

2.5 代码编译流程

2.5.1 内核编译流程

在 longan 目录下输入以下命令：

```
H616/longan$ ./build.sh config
```

```
Welcome to mkscrip setup progress
```

```
All available platform:
```

- 0. android
- 1. linux

```
Choice [android]: 0
```

```
All available ic:
```

- 0. h616

```
Choice [h616]: 0
```

```
All available board:
```

- 0. fpga
- 1. ft
- 2. p1
- 3. p2
- 4. perf1
- 5. perf1_axp152
- 6. perf2
- 7. perf3
- 8. qa
- 9. vs

```
Choice [p2]: 3
```

```
INFO: kernel defconfig: generate /home/yuguoxu/H616/longan/kernel/linux-4.9/.config by
```

```
/home/yuguoxu/H616/longan/kernel/linux-4.9/arch/arm64/configs/sun50iw9p1smp_h616_android_defconfig
```

```
*** Default configuration is based on 'sun50iw9p1smp_h616_android_defconfig'
```

```
#
```

```
# configuration written to .config
#
H616/longan$
```

输入 `.build.sh` 开始编译

```
H616SDK/longan$ ./build.sh
```

编译成功后输出内容如下：

```
regenerate rootfs cpio
15757 blocks
15758 blocks
build_ramfs
Copy boot.img to output directory ...
Copy modules to target ...

sun50iw9p1 compile Kernel successful

INFO: build kernel OK.
INFO: build rootfs ...
INFO: skip make rootfs for android
INFO: build rootfs OK.
INFO: -----
INFO: build longan OK.
INFO: -----
```

内核的代码在 `longan/linux-4.9` 目录，执行上述命令编译前会将配置文件从 `longan/kernel/linux-4.9/arch/arm64/configs/sun50iw9p1smp_h616_android_defconfig` 拷贝到 `longan/kernel/linux-4.9/.config` 作为默认配置，下次编译时可以直接在 `longan` 下运行 `./build.sh`，将继续采用上一次的 `.config` 配置

2.5.2 Android 代码编译流程

```
cd ~/workspace/AndroidQ/android
source ./build/envsetup.sh ----导入环境变量(执行者一句之后会默认选择openjdk8)
lunch ----根据自己的开发平台选择方案
cupid_p2-userdebug ----根据自己的开发平台输入对应的名字或者数字
extract-bsp ----拷贝longan下的内核和模块到android中
make -j8 ----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 ----打包生成固件
```

2.5.3 jack 编译问题

AndroidQ 使用了 Jack 编译工具, 首次编译时容易有 Jack 编译失败的问题出现, 如果对于单用户模式的开发的话, 则 Jack 服务器的端口号不需要修改, 但如果对应多用户服务器的开发模式的话, 则会出现有用用户占用了 jack 服务器的端口, 导致无法编译通过的问题出现, 需要修改默认的 jack 服务器端口。

编译到一半后, 编译系统会自动在 ~/目录 下生成 ~/.jack-settings 文件, 用 vi 打开此文件, 修改端口号

```
SERVER_PORT_SERVICE=9076 改成没有被占用的端口号
SERVER_PORT_ADMIN=9077 改成没有被占用的端口号
```

接着修改 ~/.jack-server/config.properties 文件, 如果没有这个文件, 就继续编译 Android 代码, 系统会自动生成这个文件, 等到报错之后再打开修改

```
jack.server.service.port=9076 改成与上面的端口号一致
jack.server.admin.port=9077 改成与上面的端口号一致
```

然后运行以下命令启动 jack 服务器, 如果配置没有问题的话, 下面命令

```
jack-admin start-server
```

注意 ~/.jack-server/目录下的文件, 要确保文件的属性不变, 建议使用 vi 而不要使用其他可能改

变文件属性的编辑器进行修改，否则也会导致编译错误，正常情况下的文件属性如下：

```
-rw----- 1 yuguoxu yuguoxu 2082 8月 26 2016 client.jks  
-rw----- 1 yuguoxu yuguoxu 2794 12月 6 10:18 client.pem  
-rw----- 1 yuguoxu yuguoxu 282 8月 26 2016 config.properties  
drwxrwxr-x 2 yuguoxu yuguoxu 4096 8月 26 2016 jack  
-rw----- 1 yuguoxu yuguoxu 4378061 8月 26 2016 launcher.jar  
drwx----- 2 yuguoxu yuguoxu 4096 3月 30 19:53 logs  
-rw----- 1 yuguoxu yuguoxu 4758810 8月 26 2016 server-1.jar  
-rw----- 1 yuguoxu yuguoxu 2067 8月 26 2016 server.jks  
-rw----- 1 yuguoxu yuguoxu 1042 12月 6 10:18 server.pem
```

2.5.4 刷机方法

将生成的 img 固件拷贝出来，用 PhoenixSuit 选择固件后，按住 uboot 按键，再连接 USB0 到电脑上电，再点击 PhoenixSuit 的确认按键，即可进行刷机

3. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner. The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This document neither states nor implies warranty of any kind, including fitness for any particular application. tates nor implies warranty of any kind, including fitness for any particular application.