

Android 开发手册及内容定制说明

V1.0

版本历史

Version	Date	author	Change
1.0	2016-12-28	宦飞	初稿

目录

Android 开发手册及内容定制说明	1
一、 概述.....	3
1.1 主控芯片介绍.....	3
1.2 外围设备介绍.....	3
1.3 软件资源介绍.....	3
二、 建立开发环境.....	3
2.1 硬件资源.....	3
2.2 软件资源.....	3
2.2.1 安装 JDK(linux).....	4
2.2.2 安装平台支持的软件(linux).....	4
2.2.3 安装编译工具链(linux).....	4
2.2.4 安装 PhoenixCard(windows)	4
2.2.5 其他软件(windows).....	4
三、 源码下载及编译.....	4
3.1 源码下载.....	4
3.2 目录结构.....	5
3.3 源码编译.....	5
3.4 固件打包及烧录.....	5
3.4.1 完全打包.....	5
3.4.2 部分打包.....	6
3.4.3 fastboot 更新系统	6
四、 调试与定制修改.....	7
4.1、u-boot 相关修改:.....	7
4.4.1 烧写工具链修改.....	7
4.4.2 修改 dram 配置	7
4.4.3 修改 u-boot 的环境变量.....	7
4.5 、linux 内核修改.....	7
4.6、Android 定制修改.....	8
4.7、其他相关定制修改.....	8
4.7.1 Dram 参数测试.....	8
4.7.2 不同方式启动系统.....	9
4.7.3 定制不同分辨率的显示.....	9

一、概述

本文档用于介绍基于 A20 环境的 Android 镜像的编译以及内容定制修改方法和常用模块的定制介绍。

1.1 主控芯片介绍

A20 主控室采用双核 Cortex-A7 架构的 CPU，主频达 1G，GPU 采用 Mali-400MP2，兼容性十分出色。最高支持 2160P 的视频解码和 1080P@30fps 的编码，多媒体性能优异。此芯片支持 1G 的 DDR3(L)/LPDDR2 内存。另外还支持 1024x768 或 1024x600 等多种分辨率。

1.2 外围设备介绍

本平台支持 WIFI 模块，3G 模块，TF(SD)卡扩展模块以及多种传感器。

1.3 软件资源介绍

本平台是建立在 Android4.2 基础之上的，linux 内核版本为 3.4。Android 生态系统支持影音，网络，娱乐，系统管理等丰富的扩展。

二、建立开发环境

本章节介绍 Android 开发环境所需的软硬件资源以及环境的搭建。

2.1 硬件资源

- A、开发板一套(包括核心板、底板、数据线等)
- B、两台 PC：一台作为编译服务器(本文档将以 Ubuntu12.04 为例)，另外一台用于烧写固件(window 7 系统)。
- C、串口线(方便调试)，12V 电源，SD 卡及读卡器

2.2 软件资源

Linux 主机需要 64bit 系统，本机使用 64bit 的 ubuntu12.04，硬盘空间只有 100G，一般 Linux 主机中还需要：

- A、Python 2.6 以上版本
- B、GUN Make 的 3.81-3.82 版本
- C、JDK 6
- D、Git 的 1.7 或者更高版本

Windows 7 或者 windows 8 作为固件烧写机器和本地调试环境，一般来说 window 主机需要：

- A、PhoenixCard
- B、USB 转串口驱动
- C、Android SDK 环境(ADB 环境)

2.2.1 安装 JDK(linux)

下载 JDK6 并在配置环境变量，在/etc/profile 末尾添加

```
JAVA_HOME=/usr/lib/java/jdk1.6.0_45/  
PATH=$JAVA_HOME/bin:$PATH  
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar  
export JAVA_HOME  
export PATH  
export CLASSPATH
```

2.2.2 安装平台支持的软件(linux)

根据自己 linux 版本选择，详见 <https://source.android.com/source/initializing.html>
本机使用如下：

```
$ sudo apt-get install git gnupg flex bison gperf build-essential \  
zip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev \  
libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 \  
libgl1-mesa-dev g++-multilib mingw32 tofrodos \  
python-markdown libxml2-utils xsltproc zlib1g-dev:i386  
$ sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib/i386-linux-gnu/libGL.so
```

2.2.3 安装编译工具链(linux)

下载 arm-linux-gcc 4.6 版本以及 5.4 版本并解压到/var/lib/下，编译 uboot 使用 5.4 版本，
kernel 使用的是 4.6 的较老版本。

2.2.4 安装 PhoenixCard(windows)

下载该软件并安装即可。

2.2.5 其他软件(windows)

建议安装 putty 或者 xshell，并且通过 SSH 连接到 linux 编译器进行源码的编译。

三、 源码下载及编译

本章节介绍源码下载和源码结构以及编译过程。所有源码都在 <https://bitbucket.org/>中下载(自行申请权限加入项目组)，并将 ssh-keygen 生产的密钥放入到 bitbucket 的个人账户中，否则将没有权限下载源码。

3.1 源码下载

先安装好 git 和 repo，并输入 `repo init -u git@bitbucket.org:segmetics/manifests.git -b bsp`，然后 `repo sync` 等待一段时间下载所有源码，下载完的源码目录如下图：

```
max@max:~/Workdir/A20/android/src$ ls
allwinner-tools  build      chosen_board.mk  Makefile  scripts      u-boot
android          buildenv.sh linux-3.4        -nostdinc sunxi-boards
```

3.2 目录结构

下载的目录结构如上图所示，修改根目录中的 Makefile 中的 TOOLCHAIN_4 和 TOOLCHAIN_5 中交叉编译的路径，并编辑修改 chosen_board.mk 中的配置来进行选择编译条件。

allwinner-tools: 全志提供的工具包，用于镜像 img 的制作，usb 烧写系统，sd 卡烧写系统等，勿做修改。

Scripts: 包含编译 Android 所需的脚本。

U-boot: uboot 的源码包。

Linux-3.4: linux 内核的源码包。

Android: Android 的源码包。

Sunxi-boards: 包括板级相关配置修改，主要包含管脚配置、uboot 环境变量、分区信息、sd 卡烧写配置等。

Build: 主要包括 Android, kernel, uboot 的生成目录。

Chosen_board.mk: 对不同平台的版本配置。

Makefile: u-boot、kernel、Android、livesuit 的编译选项的规则。

3.3 源码编译

首先配置好交叉编译器的路径以及 chosen_board.mk 配置(默认即可)。

编译 u-boot:

提前安装好 mkimage 等工具，然后直接 make u-boot，等待几分钟即可，如需配置编译选项，执行 make u-boot-config 即可。

编译 kernel

直接执行命令 make linux，等待几分钟即可，如需配置内核编译选项，执行 make linux-config 即可。

编译 android

直接执行命令 make android 即可，第一次编译需要等待较长一段时间。如需配置 eng, user, userdebug 版本只要在 chosen_config.mk 配置 VARIANT 即可。

3.4 固件打包及烧录

3.4.1 完全打包

直接执行命令 make livesuit，然后会在 build 目录下生成对应的 img 文件，将该文件拷贝至 windows 下，然后插入 SD 卡并打开 PhoenixCard 软件如下图：



选择镜像文件并选择对应 SD 卡的盘符，如果需要烧写开发板的 emmc，请选择卡量产并点击烧录即可。等待烧写完毕，然后将 SD 卡插入开发板，并重新给开发板上电启动，此时 SD 卡会给开发板的 emmc 烧写 img，等待烧写完毕拔出 SD 卡即可。如需从 SD 卡启动开发板，请在上图的烧写模式中选择卡启动并点击烧写，等待烧写完毕。然后在开发板中插入 SD 卡并重新上电即可从 SD 卡启动开发板。

3.4.2 部分打包

在根目录下使用命令 `make android-bsp`，即可给 Android 更新 linux 内核和模块，在编译 Android 时，可以通过下面命令进行部分编译打包：

```
$ cd android
$source build/envsetup.sh
$lunch #选择产品
$make bootimage
```

这样就生成了 boot.img，类似的方法可以重新打包生成 system.img.

3.4.3 fastboot 更新系统

fastboot 是一种线刷，就是使用 USB 数据线连接到 android 开发板的一种刷机模式，在 A20 主控中可以使用 fastboot 的功能来实现局部的系统的更新。

A、进入 fastboot 模式：

可以在 android 系统起来之后输入 `reboot bootloader`，或者在启动开发板中，在串口界面敲任意按键进入 uboot，然后输入 `fastboot` 命令进入 fastboot 模式。

B、fastboot 命令使用：

在配好环境变量的 windows 命令行使用 fastboot 命令。

擦除分区命令：

```
$fastboot erase boot #擦除 boot 分区
$fastboot erase system #擦除 system 分区
$fastboot erase data #擦除 data 分区
```

烧写分区命令：

```
$fastboot flash boot boot.img #把 boot.img 烧写到 boot 分区
$fastboot flash system system.img #把 system.img 烧到 system 分区
$fastboot flash data userdata.img #把 userdata.img 烧到 data 分区
```

四、 调试与定制修改

本章节主要介绍 uboot、kernel、android 的定制与修改方法。针对不同的需求来修改源码以满足项目的要求。

4.1、u-boot 相关修改：

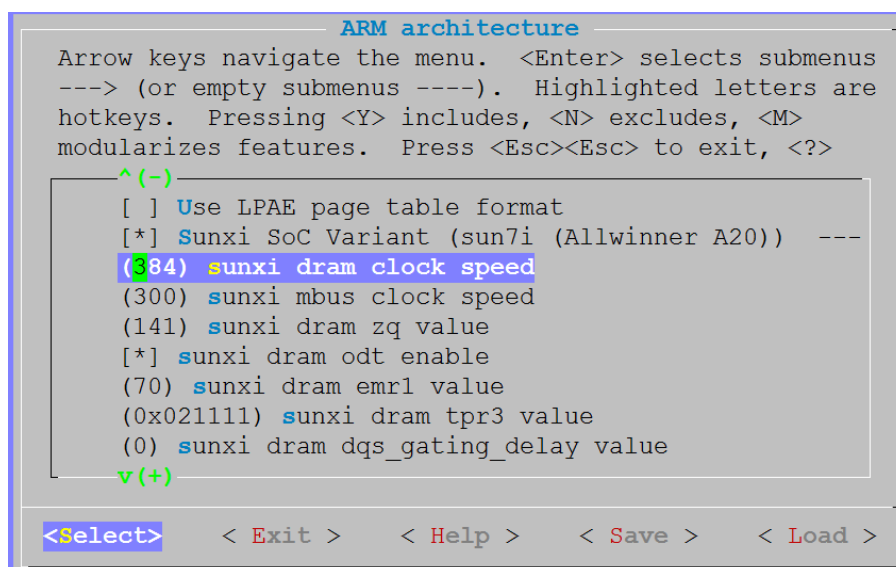
由于配置 dram 的相关参数，平台的 u-boot 使用 mainline 的源码。

4.4.1 烧写工具链修改

在 u-boot/sprite 目录中，sprite.c 中有烧写流程的相关代码，如需修改流程只要根据需求修改对应的代码。

4.4.2 修改 dram 配置

dram 配置是在 spl 阶段，所以可以通过 make u-boot-config 来配置 dram 参数。如下图：



然后进行参数设置即可。

4.4.3 修改 u-boot 的环境变量

编辑 sunxi-boards/sun7i_soc/mxBenjen_board/android-sys_eng/env.cfg，按照语法规则配置相关的环境变量。

4.5 、linux 内核修改

根据不同模块的管脚使用可以在 unxi-boards/sun7i_soc/mxBenjen_board/android-sys_eng/sys_config.fex 中修改相关的配置。针对其他的模块的修改与配置可以参考全志提供的文档《A20_系统配置手册》。

4.6、Android 定制修改

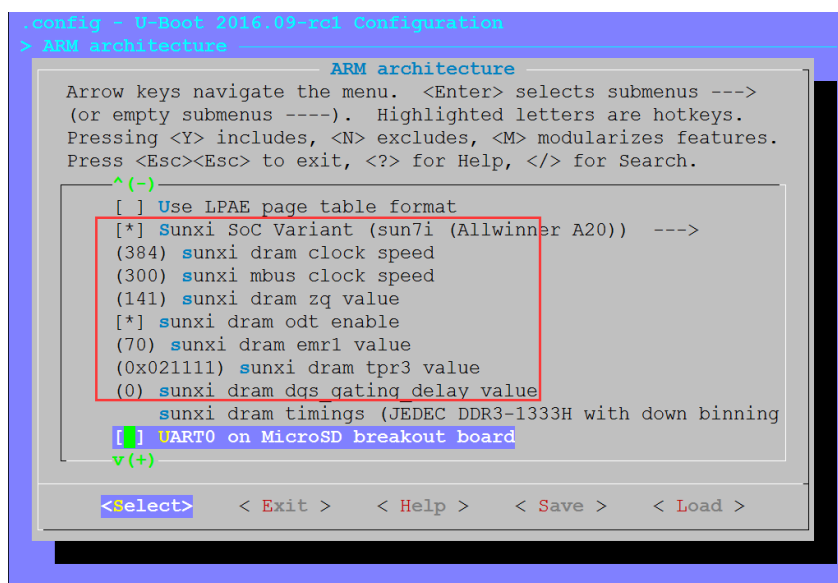
根据全志提供的文档《A20_Android 内容定制文档》来进行相对应的修改。

4.7、其他相关定制修改

4.7.1 Dram 参数测试

使用 mainline 的 uboot、sunxi 的 linux 内核。将 uboot 烧入核心板的 flash，然后通过 NFS 启动内核和 rootfs(方便查看测试结果)，具体如下：

A、make u-boot-config 配置相关 dram 的参数



B、配置 uboot 的环境变量，如下：

```
$env set machid 0x0f35;
$env set autoload no
$env set bootm_boot_mode sec
$env set ipaddr2 "::::mxB:eth1:dhcp"
$env set serverip 10.9.8.7
$env set nfsroot 10.9.8.7:/volume1/linux_rootfs;
$env set bootargs "console=ttyS0,115200 root=/dev/nfs rw nfsroot=${nfsroot}
ip=${ipaddr2} loglevel=4 rootwait init=/root/dram/memtest.sh"
$env set bootcmd "usb reset;dhcp;nfs 0x43000000 ${nfsroot}/sys_config_vga.bin;nfs
0x40007800 ${nfsroot}/ulimage_new_board;bootm 0x40007800;"
```

C、做好支持 NFS 的 kernel 后放到服务器上环境变量配置 的参数,然后启动板子进行测试,经过长时间大量测试后可以通过命令 ./a10-tpr3-html-report /var/tpr3_results > /var/tpr3_results/report.html 查看结果,使用浏览器打开 report.html 文件,通过绿色形状来判断,若都是红色或其他不合理情况,修改 u-boot 参数来继续测试。详细可见

http://linux-sunxi.org/A10_DRAM_Controller_Calibration#Overview_of_the_DRAM_controller_features_affecting_the_clock_speed_limit_and_reliability。

4.7.2 不同方式启动系统

开发板可以从 emmc 或者 sd 卡启动 uboot(优先从 SD 卡启动)，然后可以根据 uboot 的环境变量设置可以从 SD 卡、emmc、NFS 启动 kernel 和 rootfs，可以根据不同应用场景灵活运用。

4.7.3 定制不同分辨率的显示

开发板支持 lvds, vga, hdmi 多种显示方式。下面介绍 vga 分辨率和 lvds 的相关配置，修改完成配置清空编译目录重新编译，否则会出现修改不生效现象。

A、vga 分辨率

vga 分辨率需要修改对面平台下面 device/softwinner/*/*.mk 文件，将 persist.sw.display.mode 和 persist.sw.display.format 进行修改同时将 fex 配置文件中的 screen0_output_mode 进行配置，注意若为 VGA 配置时，对应参数值如下：

```
DISP_VGA_H1680_V1050    = 0,
DISP_VGA_H1440_V900     = 1,
DISP_VGA_H1360_V768     = 2,
DISP_VGA_H1280_V1024    = 3,
DISP_VGA_H1024_V768     = 4,
DISP_VGA_H800_V600      = 5,
DISP_VGA_H640_V480      = 6,
DISP_VGA_H1440_V900_RB  = 7, //not support yet
DISP_VGA_H1680_V1050_RB = 8, //not support yet
DISP_VGA_H1920_V1080_RB = 9,
DISP_VGA_H1920_V1080    = 0xa,
DISP_VGA_H1280_V720     = 0xb,
DISP_VGA_MODE_NUM       = 0xc,
```

B、lvds 配置修改

根据显示器的参数进行一一配置 fex 文件，然后需要修改对面平台下面 device/softwinner/*/*.mk 文件，将 persist.sw.display.mode 和 persist.sw.display.format 属性进行配置。注意在配置 fex 文件时需要将其中 lcd_vbp, lcd_vt, lcd_y 需要满足如下关系式，否则驱动报错（lcd_vt 参数可能与实际显示器参数存在不匹配现象，以公式计算来修改 lcd_vt 为准）。

```
"(lcd_vt/2) >= (lcd_vbp+lcd_y+2)"
```