CUBIEBOARD
http://cubieboard.org

CubieAIO Android Building Guide

CubieAIO

Website:http://cubieboard.org/
Email: support@cubietech.com

Version	Author	Modification	Check
V-1.0	Andy	Init version	Aaron

Table of Contents

1.Introduction....4
2.Overview....4
3.Software requirements....4
3.1.Install JDK....4
3.2.Compile the required package of installation (Ubuntu 12.04) 5
4.Download SDK....5
5.Building....6
5.1. Building Lichee 6
5.2.The overall compilation 7

1. Introduction

CubieAIO is based on product of Einteins A20 , Cubietech customized a android4.4.2 os based on A20 SDK , and open the source code to the community .

2. Overview

This is a quick guide for CubieAIO SDK :

- How to download SDK
- How to set up PC compilation environment
- How to build and pack a image

3. Software requirements

Developing host PC with Linux os , Ubuntu12.04 AMD64 is suggested

- USB upgrading tools : Livesuit or PhoenixSuit
- Cross-compilation environment

3.1. Install JDK

```
$ wget dl.cubieboard.org/software/tools/android/jdk1.6.0_45.tar.gz
```

Unpack :

```
$sudo tar -zxvf jdk1.6.0_45.tar.gz
$ sudo vim ~/.bashrc
```

Add :

```
JAVA_HOME=/jdk_path
export JRE_HOME=/jdk_path/jre
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
```

jdk-paththe: path of Download unpacked JDK

Which saved enforced and verify success:

```
$source ~/.bashrc&&java -version
java version "1.6.0_45"
Java(TM) SE Runtime Environment (build 1.6.0_45-b06)
Java HotSpot(TM) 64-Bit Server VM (build 20.45-b01, mixed mode)
```

3.2. Compile the required package of installation (Ubuntu 12.04)

```
$sudo apt-get update
$sudo apt-get upgrade
$sudo apt-get install build-essential u-boot-tools uboot-mkimage binutils-arm-linux-gnueabihf
$sudo apt-get install gcc g++ gcc-arm-linux-gnueabi gcc-arm-linux-gnueabihf g++-multilib
$sudo apt-get install cpp-arm-linux-gnueabihf libusb-1.0-0 libusb-1.0-0-dev wget fakeroot
$sudo apt-get install kernel-package zlib1g-dev libncurses5-dev build-essential
$sudo apt-get install texinfo texlive ccache zlib1g-dev gawk bison flex gettext uuid-dev
$sudo apt-get install ia32-libs git gnupg flex bison gperf build-essential zip
$sudo apt-get install curl libc6-dev x11proto-core-dev libx11-dev:i386 lib32ncurses5-dev
$sudo apt-get install libreadline6-dev:i386 mingw32 tofrodos python-markdown
$sudo apt-get install libxml2-utils xsltproc zlib1g-dev:i386 libgl1-mesa-dev
```

4. Download SDK

You can download the sdk from here: CubieAIO/Image/Android/V2.0/

A few simple steps to get your source :

```
$ sudo apt-get install git
$ mkdir -p  a20/lichee
$ mkdir -p a20/android

$ cd a20/lichee
$ tar -zxpf a20_homelet_v2.0_lichee.git.tar.gz
$ git reset --hard

$ cd ../../a20/android
$ tar -zxpf a20_homelet_v2.0_android4.2.git.tar.gz
$ git reset --hard
```

First I think you should switching branch, this source has two branches, one is Cubieboard2, the other is Einstein，we should switch to Einstein.

$ git branch -a

$ git checkout Einstein

5. Building

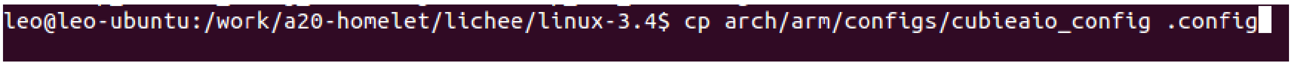
It is two directory after extract the source code package, one is the android ,another lichee, kernel compilation run in the lichee. Lichee mainly compile some module of equipment , the drivers, the kernel, and so on, and configuration files. Compiled modules and the kernel will copy when compile the whole android.

5.1. Building Lichee

The step of compilation as follows ：

1.**Copy the configuration file of kernel**

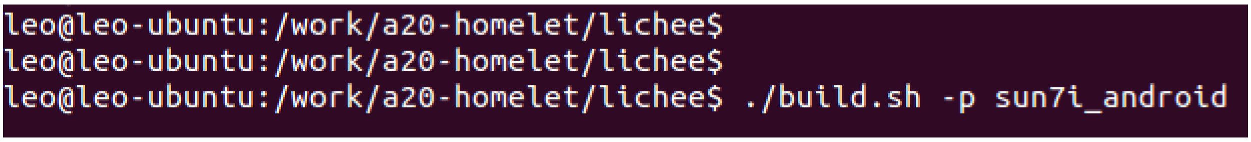
$ cp arch/arm/configs/cubieaio_config .config



```
leo@leo-ubuntu:/work/a20-homelet/lichee/linux-3.4$ cp arch/arm/configs/cubieaio_config .config
```

2.**Compile the kernel**

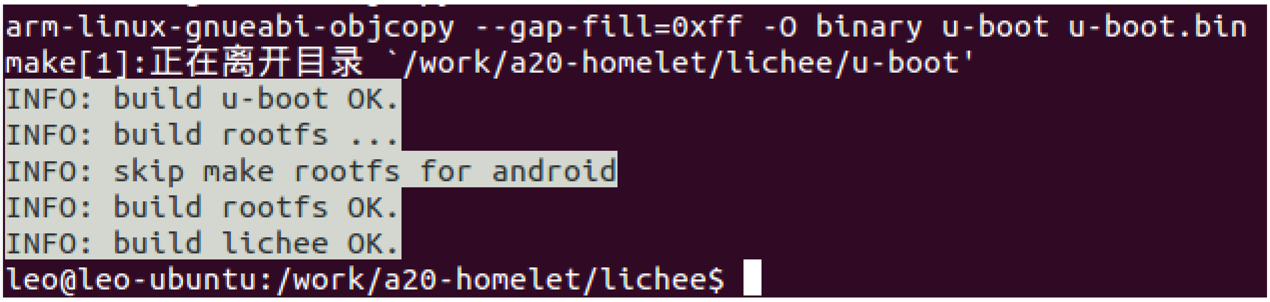
$ ./build.sh -p sun7i_android



```
leo@leo-ubuntu:/work/a20-homelet/lichee$
leo@leo-ubuntu:/work/a20-homelet/lichee$
leo@leo-ubuntu:/work/a20-homelet/lichee$ ./build.sh -p sun7i_android
```

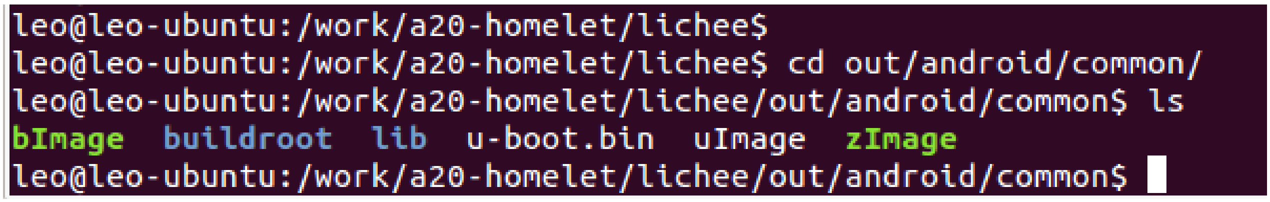
waitting..........

completed:



```
arm-linux-gnueabi-objcopy --gap-fill=0xff -O binary u-boot u-boot.bin
make[1]:正在离开目录 `/work/a20-homelet/lichee/u-boot'
INFO: build u-boot OK.
INFO: build rootfs ...
INFO: skip make rootfs for android
INFO: build rootfs OK.
INFO: build lichee OK.
leo@leo-ubuntu:/work/a20-homelet/lichee$
```

You can enter the lichee/out after Compile and can see the product of the compilation.



```
leo@leo-ubuntu:/work/a20-homelet/lichee$
leo@leo-ubuntu:/work/a20-homelet/lichee$ cd out/android/common/
leo@leo-ubuntu:/work/a20-homelet/lichee/out/android/common$ ls
bImage  buildroot  lib  u-boot.bin  uImage  zImage
leo@leo-ubuntu:/work/a20-homelet/lichee/out/android/common$
```

5.2. The overall compilation

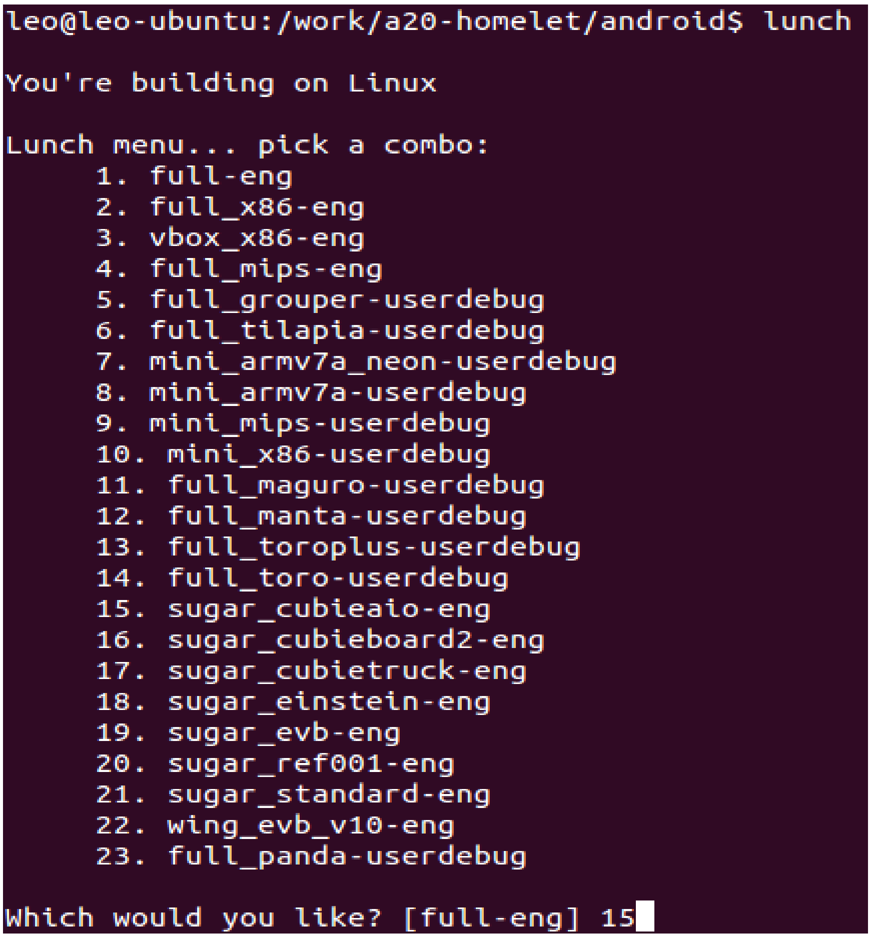
After the kernel compiled, into the android directory. Here to perform the firmware compile, generating system. Img, userdata. Img image, finally they are packaged to firmware as we need.

Steps are as follows:

leo@leo-ubuntu:/work/a20-homelet$ cd android

leo@leo-ubuntu:/work/a20-homelet/android$ source build/envsetup.sh

leo@leo-ubuntu:/work/a20-homelet/android$ lunch



```
leo@leo-ubuntu:/work/a20-homelet/android$ lunch

You're building on Linux

Lunch menu... pick a combo:
     1. full-eng
     2. full_x86-eng
     3. vbox_x86-eng
     4. full_mips-eng
     5. full_grouper-userdebug
     6. full_tilapia-userdebug
     7. mini_armv7a_neon-userdebug
     8. mini_armv7a-userdebug
     9. mini_mips-userdebug
     10. mini_x86-userdebug
     11. full_maguro-userdebug
     12. full_manta-userdebug
     13. full_toroplus-userdebug
     14. full_toro-userdebug
     15. sugar_cubieaio-eng
     16. sugar_cubieboard2-eng
     17. sugar_cubietruck-eng
     18. sugar_einstein-eng
     19. sugar_evb-eng
     20. sugar_ref001-eng
     21. sugar_standard-eng
     22. wing_evb_v10-eng
     23. full_panda-userdebug

Which would you like? [full-eng] 15
```

input 15 (choose sugar_cubieaio-eng)

If the number does not match with the English, please select the number before “sugar_cubieaio-eng” or input “sugar_cubieaio-eng”。

Copy the kernel and modules:

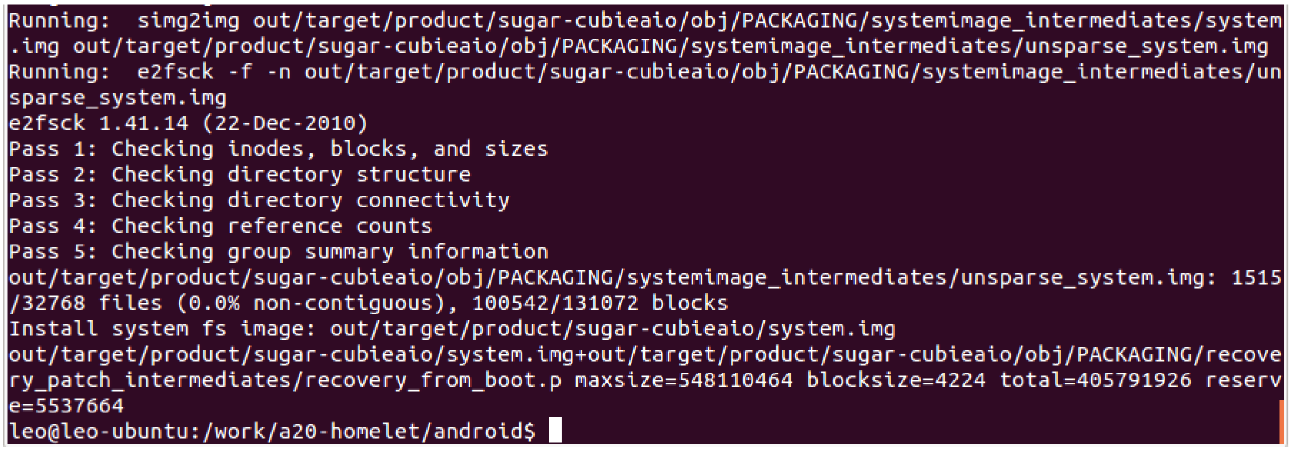
```
leo@leo-ubuntu:/work/a20-homelet/android$ extract-bsp
```

Compile:

```
leo@leo-ubuntu:/work/a20-homelet/android$ make -j8
```

waitting...........

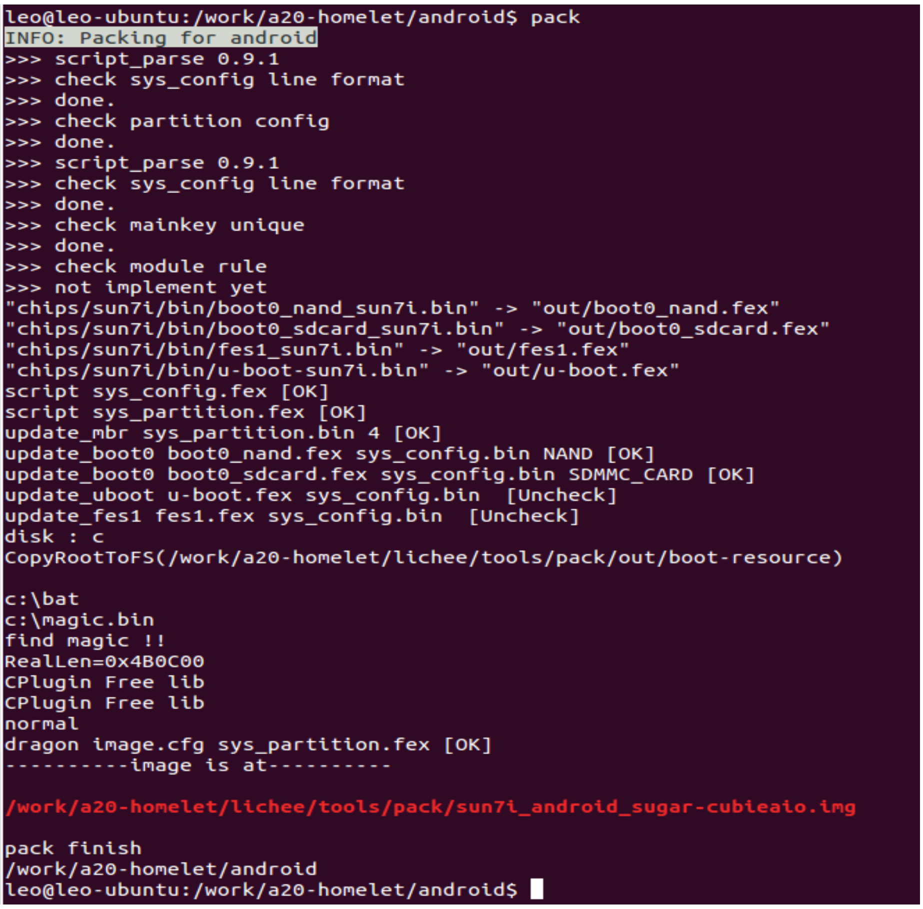
completed:



```
Running:  simg2img out/target/product/sugar-cubieaio/obj/PACKAGING/systemimage_intermediates/system
.img out/target/product/sugar-cubieaio/obj/PACKAGING/systemimage_intermediates/unsparse_system.img
Running:  e2fsck -f -n out/target/product/sugar-cubieaio/obj/PACKAGING/systemimage_intermediates/un
sparse_system.img
e2fsck 1.41.14 (22-Dec-2010)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
out/target/product/sugar-cubieaio/obj/PACKAGING/systemimage_intermediates/unsparse_system.img: 1515
/32768 files (0.0% non-contiguous), 100542/131072 blocks
Install system fs image: out/target/product/sugar-cubieaio/system.img
out/target/product/sugar-cubieaio/system.img+out/target/product/sugar-cubieaio/obj/PACKAGING/recove
ry_patch_intermediates/recovery_from_boot.p maxsize=548110464 blocksize=4224 total=405791926 reserv
e=5537664
leo@leo-ubuntu:/work/a20-homelet/android$
```

pack after completed:

```
leo@leo-ubuntu:/work/a20-homelet/android$ pack
```



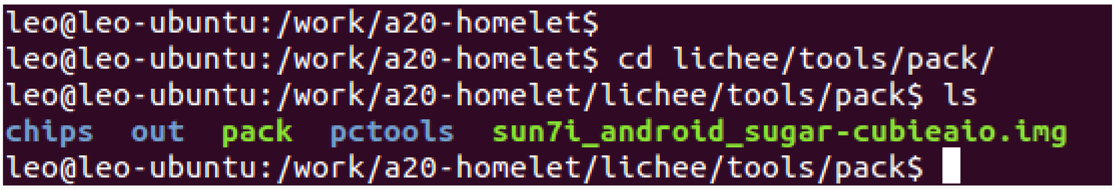
```
leo@leo-ubuntu:/work/a20-homelet/android$ pack
INFO: Packing for android
>>> script_parse 0.9.1
>>> check sys_config line format
>>> done.
>>> check partition config
>>> done.
>>> script_parse 0.9.1
>>> check sys_config line format
>>> done.
>>> check mainkey unique
>>> done.
>>> check module rule
>>> not implement yet
"chips/sun7i/bin/boot0_nand_sun7i.bin" -> "out/boot0_nand.fex"
"chips/sun7i/bin/boot0_sdcard_sun7i.bin" -> "out/boot0_sdcard.fex"
"chips/sun7i/bin/fes1_sun7i.bin" -> "out/fes1.fex"
"chips/sun7i/bin/u-boot-sun7i.bin" -> "out/u-boot.fex"
script sys_config.fex [OK]
script sys_partition.fex [OK]
update_mbr sys_partition.bin 4 [OK]
update_boot0 boot0_nand.fex sys_config.bin NAND [OK]
update_boot0 boot0_sdcard.fex sys_config.bin SDMMC_CARD [OK]
update_uboot u-boot.fex sys_config.bin  [Uncheck]
update_fes1 fes1.fex sys_config.bin  [Uncheck]
disk : c
CopyRootToFS(/work/a20-homelet/lichee/tools/pack/out/boot-resource)

c:\bat
c:\magic.bin
find magic !!
RealLen=0x4B0C00
CPlugin Free lib
CPlugin Free lib
normal
dragon image.cfg sys_partition.fex [OK]
----------image is at----------

/work/a20-homelet/lichee/tools/pack/sun7i_android_sugar-cubieaio.img

pack finish
/work/a20-homelet/android
leo@leo-ubuntu:/work/a20-homelet/android$ 
```

The firmware will be generated in the lichee/tools/pack .



```
leo@leo-ubuntu:/work/a20-homelet$
leo@leo-ubuntu:/work/a20-homelet$ cd lichee/tools/pack/
leo@leo-ubuntu:/work/a20-homelet/lichee/tools/pack$ ls
chips  out  pack  pctools  sun7i_android_sugar-cubieaio.img
leo@leo-ubuntu:/work/a20-homelet/lichee/tools/pack$ 
```

The firmware's path:

lichee/tools/pack/sun7i_android_sugar-cubieaio.img