注册 | 登录   Home page

搜索

front page   Featured section   Enterprise Zone   Electronic DIY   active channel

latest post   Technology SOS   check in

Opportunities are waiting for you who have the courage to recommend yourself in the collection of candidates for the Light of the Times series, which has influenced China's electro

Electronics World » Forum Homepage » DIY and Open Source Design » Open Source Hardware » SINA33-V1.0 android system compilation process

Post a new post

Daily check-in

The compilation process of SINA33-V1.0 android system

Xinlinsi FAE   helper   2016-05-17 16:44:49   Reward

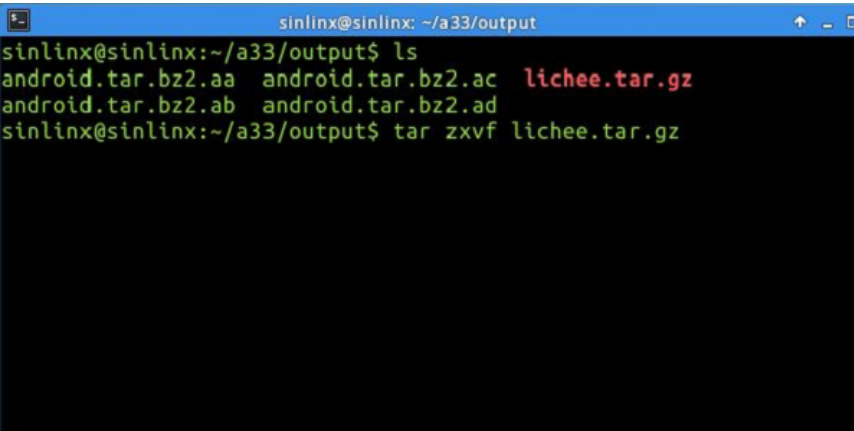Just look at the first floor of the landlord

SINA33-V1.0 compile Android system

(1) Unzip the Android source code

1.lichee : source code location: CD \ source code \lichee.tar.gz

Copy it to your own working directory and decompress it

tar zxvf lichee.tar.gz

2. Android : source code location: CD \ source code \android.tar.bz2.a*

Copy it to your own working directory and decompress it

Cat android.tar.bz2.a*|tar xj

Top Categories

ST	MCU
STM32	MCU
Communications and Wireless Technology	IoT technology
DIY	
Electronic DIY	Board Trial
basic knowledge	Software and Operating System
i love life	small e cafeteria

3. The lichee directory is the source code of uboot , linux and some script configurations, and the android directory is the source code of the android part. These two parts need to be compiled separately.

(2) Compile the lichee directory

1. Enter the lichee directory

cd lichee

2. Execute the configuration command before the first compilation, and the subsequent recompilation can be omitted.

./build.sh config

Select 0 sun8iw5p1 Enter

Select 0 android and press Enter

Select 0 linux-3.4 and press Enter

Select 4 y3 Enter

3. Compile the lichee directory

./build.sh

4. The compilation is complete

```
modules/mali'
regenerate rootfs cpio
8772 blocks
9475 blocks
build_ramfs
Copy boot.img to output directory ...
Copy modules to target ...
INFO: build kernel OK.
INFO: build rootfs ...
INFO: skip make rootfs for android
INFO: build rootfs OK.
INFO: ----------------------------------
INFO: build lichee OK.
INFO: ----------------------------------
sinlinx@sinlinx:~/a33/output/lichee$
```

(3) Compile the android directory

    1. Enter the android directory

        cd android

    2. Set environment variables

        Source build/envsetup.sh


```
sinlinx@sinlinx:~/a33/output/android$ source build/envsetup.sh
including device/asus/tilapia/vendorsetup.sh
including device/asus/grouper/vendorsetup.sh
including device/asus/deb/vendorsetup.sh
including device/asus/flo/vendorsetup.sh
including device/softwinner/polaris-common/vendorsetup.sh
including device/softwinner/astar-y3/vendorsetup.sh
including device/samsung/manta/vendorsetup.sh
including device/lge/hammerhead/vendorsetup.sh
including device/lge/mako/vendorsetup.sh
including device/generic/x86/vendorsetup.sh
including device/generic/mips/vendorsetup.sh
including device/generic/armv7-a-neon/vendorsetup.sh
including sdk/bash_completion/adb.bash
sinlinx@sinlinx:~/a33/output/android$
```

    3. Select configuration scheme

        Lunch

        Enter the configuration scheme we choose

        Select 9 Enter


```
sinlinx@sinlinx:~/a33/output/android$ lunch
```

```
                    sinlinx@sinlinx: ~/a33/output/android
  4.  vbox_x86-eng
  5.  aosp_tilapia-userdebug
  6.  aosp_grouper-userdebug
  7.  aosp_deb-userdebug
  8.  aosp_flo-userdebug
  9.  astar_y3-eng
 10.  astar_y3-user
 11.  aosp_manta-userdebug
 12.  aosp_hammerhead-userdebug
 13.  aosp_mako-userdebug
 14.  mini_x86-userdebug
 15.  mini_mips-userdebug
 16.  mini_armv7a_neon-userdebug

Which would you like? [aosp_arm-eng] 9
```

```
                    sinlinx@sinlinx: ~/a33/output/android
TARGET_BUILD_TYPE=release
TARGET_BUILD_APPS=
TARGET_ARCH=arm
TARGET_ARCH_VARIANT=armv7-a-neon
TARGET_CPU_VARIANT=cortex-a7
HOST_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-3.13.0-24-generic-x86_64-with-Ubuntu-14.04-tru
ty
HOST_BUILD_TYPE=release
BUILD_ID=KVT49L
OUT_DIR=out
============================================

sinlinx@sinlinx:~/a33/output/android$
```

4. Copy the compiled uboot and kernel in the lichee directory

   extract-bsp

```
                    sinlinx@sinlinx: ~/a33/output/android
sinlinx@sinlinx:~/a33/output/android$ extract-bsp
/home/sinlinx/a33/output/android/device/*/astar-y3/bImage copied!
/home/sinlinx/a33/output/android/device/*/astar-y3/modules copied
sinlinx@sinlinx:~/a33/output/android$
```

   5. Compile (j8 compiles with 8 threads, please select the number of thr
eads according to the CPU capacity of your PC)

   make -j8

```
Blocks per group: 32768
Inodes per group: 8192
Inode size: 256
Journal blocks: 3072
Label:
Blocks: 196608
Block groups: 6
Reserved block group size: 47
Created filesystem with 1669/49152 inodes and 143149/196608 blocks
+ '[' 0 -ne 0 ']'
Install system fs image: out/target/product/astar-y3/system.img
out/target/product/astar-y3/system.img+out/target/product/astar-y3
obj/PACKAGING/recovery_patch_intermediates/recovery_from_boot.p ma
size=822163584 blocksize=4224 total=577593476 reserve=8308608
sinlinx@sinlinx:~/a33/output/android$
```

6. Package after compiling

    pack


```
Vboot.fex Len: 0x4
system.fex Len: 0x224a4bdc
Vsystem.fex Len: 0x4
recovery.fex Len: 0xdb3800
Vrecovery.fex Len: 0x4
diskfs.fex Len: 0x200
Vdiskfs.fex Len: 0x4
BuildImg 0
Dragon execute image.cfg SUCCESS !
---------image is at---------

/home/sinlinx/a33/output/lichee/tools/pack/sun8iw5p1_android_y3.im

pack finish
sinlinx@sinlinx:~/a33/output/android$
```

    Finally generate the image sun8iw5p1_android_y3.img that can be burned

    The directory where the image is located is lichee/tools/pack

reply   collect   share

Updated｜What you care about about ADI

A total of 1 1/1 [1]   Jump to [   ] Page

reply

Anonymous cannot post! Please [login and register] first

Reply Ctrl+Enter