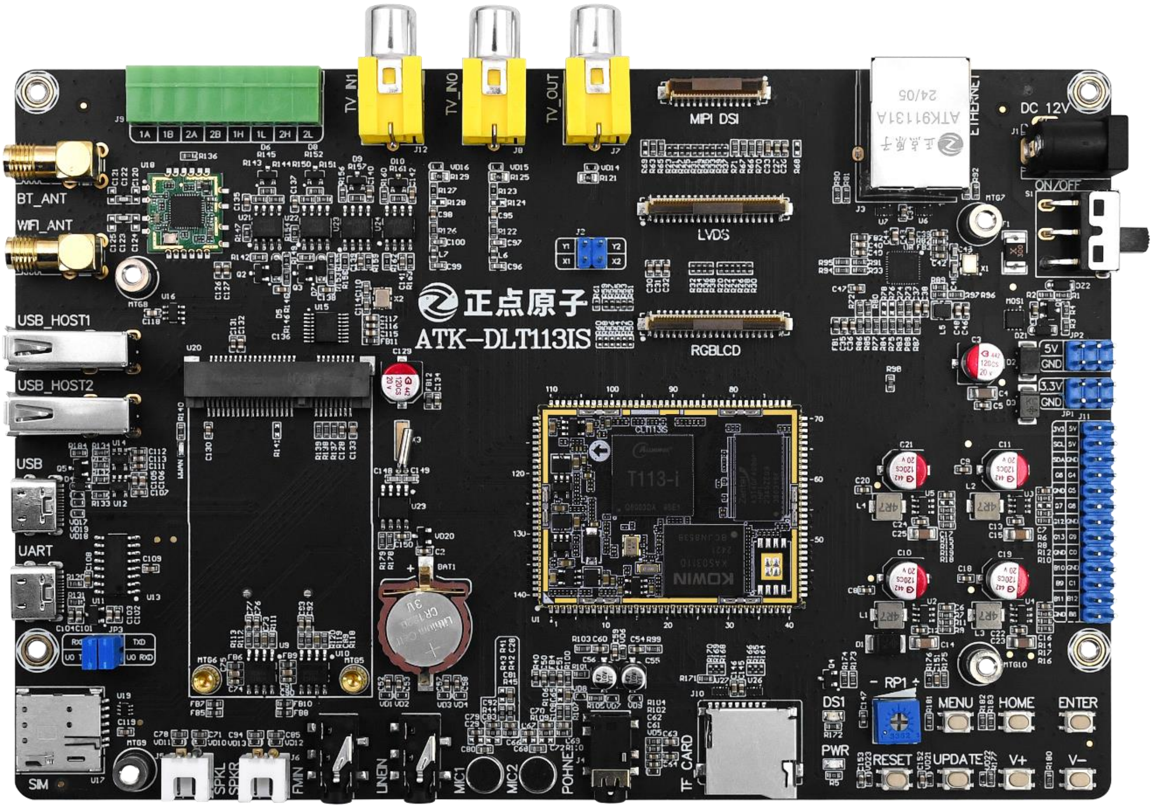
ATK-DLT113IS

Factory System SDK User Manual

V1.1



ALIENTEK

1. Shopping:

TMALL：https://zhengdianyuanzi.tmall.com

TAOBAO：https://openedv.taobao.com

2. Download

Address：http://www.openedv.com/docs/index.html

3. FAE

Website ： www.alientek.com

Forum ： http://www.openedv.com/forum.php

Videos ： www.yuanzige.com

Fax ： +86 - 20 - 36773971

Phone ： +86 - 20 - 38271790

Disclaimer

The product specifications and instructions mentioned in this document are for reference only and subject to update without prior notice; Unless otherwise agreed, this document is intended as a product guide only, and none of the representations made herein constitutes a warranty of any kind. The copyright of this document belongs to Guangzhou Xingyi Electronic Technology Co., LTD. Without the written permission of the company, any unit or individual shall not be used for profit-making purposes in any way of dissemination.

In order to get the latest version of product information, please regularly visit the download center or contact the customer service of Taobao ALIENTEK flagship store. Thank you for your tolerance and support.

Revision History:

Version	Version Update Notes	Responsible person	Proofreading	Date
V1.0	release officially	ALIENTEK	ALIENTEK	2024.04.30

Catalogue

Chapter 1. The use of SDK

The ATK-DLT113IS development board from ALIENTEK is specifically developed based on the Allwinner T113_Tina5.0-V1.0 software development kit (SDK) version. It is important to note that each SDK version of Allwinner, such as T113_Tina5.0-V1.0, has significant differences in usage, providing developers with flexible space to add commands and scripts according to their specific needs.

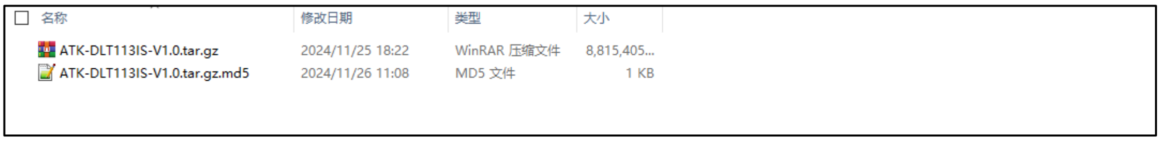
There are many things that must be done this way. Don't ask why it has to be done this way. Even if you ask, I still can't give you an answer.

There are many things that must be done this way. Don't ask why it has to be done this way. Even if you ask, I still can't give you an answer.

There are many things that must be done this way. Don't ask why it has to be done this way. Even if you ask, I still can't give you an answer.

1.1 Unzipping the Source Code

The source code path of the SDK: Development board CD-ROM A disk - Basic Materials → 1_codes → 1_SDK_source_code. As shown in the following figure:

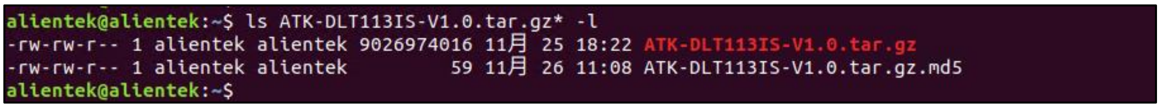


□ 名称	修改日期	类型	大小
ATK-DLT113IS-V1.0.tar.gz	2024/11/25 18:22	WinRAR 压缩文件	8,815,405...
ATK-DLT113IS-V1.0.tar.gz.md5	2024/11/26 11:08	MD5 文件	1 KB

Figure 1.1-1 The source code diagram of ATK-DLT113IS

The above picture shows two files: ATK-DLT113IS-V1.0.tar.gz and ATK-DLT113IS-V1.0.tar.gz.md5. Among them, ATK-DLT113IS-V1.0.tar.gz is a source code compressed file, which contains the source code of the project; while ATK-DLT113IS-V1.0.tar.gz.md5 is a MD5 checksum file corresponding to this source code compressed package, used to verify the integrity and accuracy of the compressed package. Please note that the V1.0 here represents the version number, and the actual version to be used should be confirmed according to the source code version you download.

Copy these two files to the Ubuntu virtual machine. Here, the author copies them to the home directory as shown in the following picture:



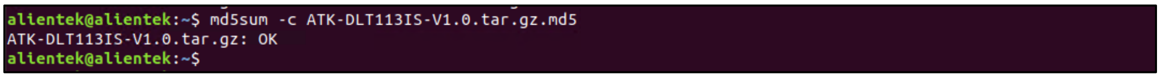
```
alientek@alientek:~$ ls ATK-DLT113IS-V1.0.tar.gz* -l
-rw-rw-r-- 1 alientek alientek 9026974016 11月 25 18:22 ATK-DLT113IS-V1.0.tar.gz
-rw-rw-r-- 1 alientek alientek         59 11月 26 11:08 ATK-DLT113IS-V1.0.tar.gz.md5
alientek@alientek:~$
```

Figure 1.1-2 Source code package copy of Ubuntu system diagram

Here, the author copies the files to the home directory, opens the terminal. Run the following command for MD5 verification (given that the source code files are large in size, to ensure the integrity of the data during the transmission process, we have specially added a file verification mechanism. Once the verification result shows failure, it indicates that the file may have been damaged during the transmission process. At this time, we recommend that you re-download the file to ensure that you obtain a complete and undamaged source code):

```
md5sum -c ATK-DLT113IS-V1.0.tar.gz.md5
```

Note: The above command requires that the MD5 checksum and the source code file must be in the same directory.

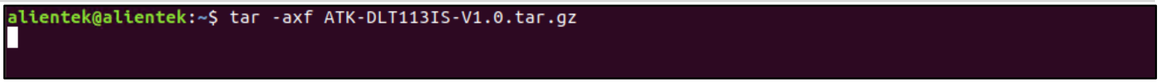


```
alientek@alientek:~$ md5sum -c ATK-DLT113IS-V1.0.tar.gz.md5
ATK-DLT113IS-V1.0.tar.gz: OK
alientek@alientek:~$
```

Figure 1.1-3 Verify the source code file

As can be seen from Figure 1.1.3, if the verification is successful, it will print "OK". After the verification is successful, run the following code to perform the decompression:

```
tar -axf ATK-DLT113IS-V1.0.tar.gz
```



```
alientek@alientek:~$ tar -axf ATK-DLT113IS-V1.0.tar.gz
```

Figure 1.1-4 Extract the source code package

1.2 Explanation of the source code directory

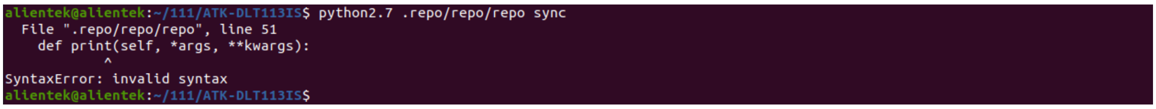
If you are using your own virtual machine, the source code compilation using ATK-DLT113IS requires the installation of the following software:

```
sudo apt install net-tools openssh-server git make gcc flex bison libssl-dev build-essential lib32z1-dev   libncurses5-dev gawk
```

In Section 1.1, we have successfully decompressed the source code package. Then, in the directory where the decompressed source code files are located, we opened the terminal. Since this source code is managed through the repo tool, we need to execute specific commands to check out (i.e., synchronize or download) the source code. Run the following command to check out:

```
.repo/repo/repo sync -l
python3 .repo/repo/repo sync -l
```

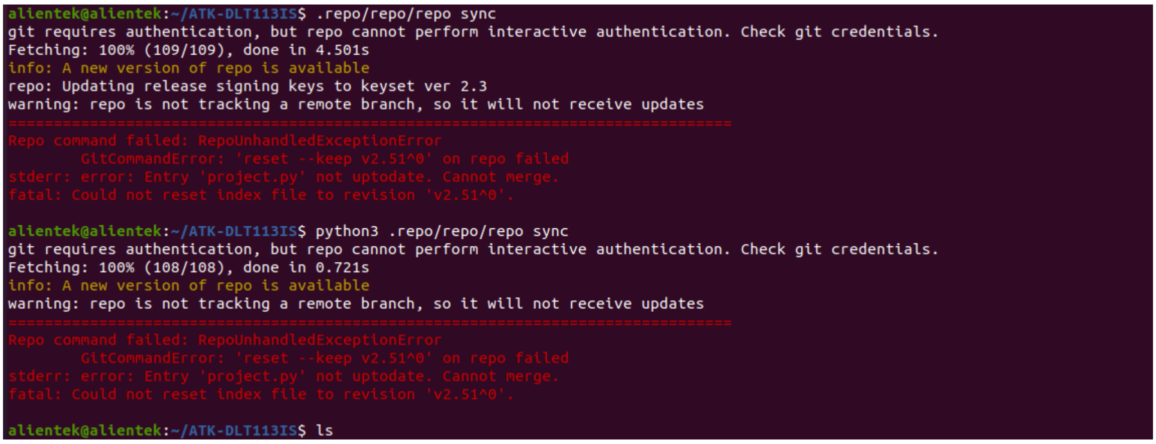
Note: Among the two commands provided above, you only need to select one to execute. If you encounter Python-related errors when running the ".repo/repo/repo sync" command, this is usually caused by incompatible Python versions. To solve this problem, you can explicitly specify to use the Python3 version for execution before running the command. If it doesn't work, please switch to a different version of Python3 (recommending python3.8). Another error is related to the repo version.



```
alientek@alientek:~/111/ATK-DLT113IS$ python2.7 .repo/repo/repo sync
  File ".repo/repo/repo", line 51
    def print(self, *args, **kwargs):
            ^
SyntaxError: invalid syntax
alientek@alientek:~/111/ATK-DLT113IS$
```

Figure 1.2-1 Incorrect Python version

If our default Python version points to the python2.x version, the error shown in Figure 1.2.1 will occur. We only need to use the python3 version for the checkout.

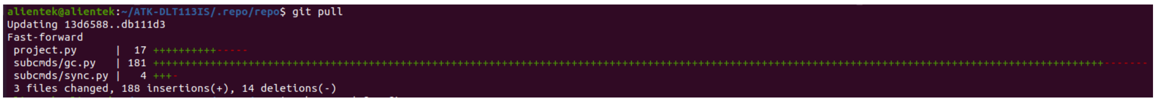


```
alientek@alientek:~/ATK-DLT113IS$ .repo/repo/repo sync
git requires authentication, but repo cannot perform interactive authentication. Check git credentials.
Fetching: 100% (109/109), done in 4.501s
info: A new version of repo is available
repo: Updating release signing keys to keyset ver 2.3
warning: repo is not tracking a remote branch, so it will not receive updates
================================================================================
Repo command failed: RepoUnhandledExceptionError
        GitCommandError: 'reset --keep v2.51^0' on repo failed
stderr: error: Entry 'project.py' not uptodate. Cannot merge.
fatal: Could not reset index file to revision 'v2.51^0'.

alientek@alientek:~/ATK-DLT113IS$ python3 .repo/repo/repo sync
git requires authentication, but repo cannot perform interactive authentication. Check git credentials.
Fetching: 100% (108/108), done in 0.721s
info: A new version of repo is available
warning: repo is not tracking a remote branch, so it will not receive updates
================================================================================
Repo command failed: RepoUnhandledExceptionError
        GitCommandError: 'reset --keep v2.51^0' on repo failed
stderr: error: Entry 'project.py' not uptodate. Cannot merge.
fatal: Could not reset index file to revision 'v2.51^0'.

alientek@alientek:~/ATK-DLT113IS$ ls
```

Figure 1.2-2 Repo version issue

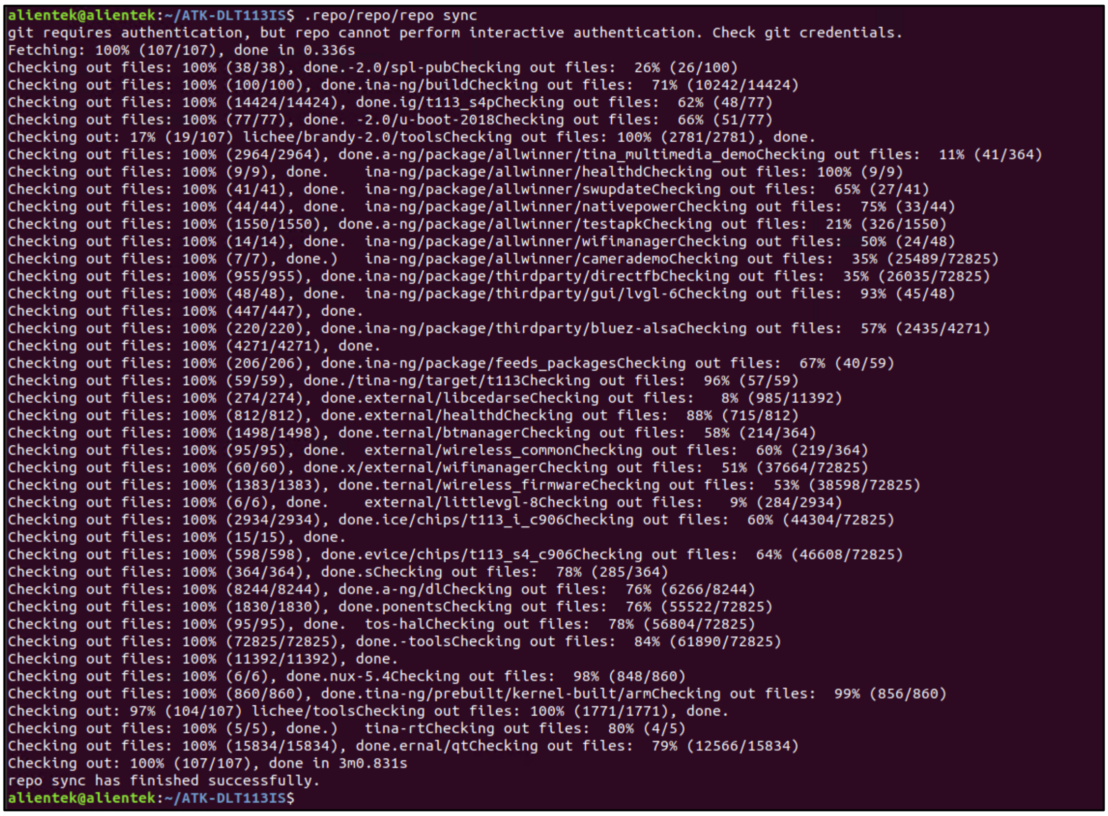
If the problem shown in Figure 1.2.2 occurs, the following command can be run to resolve it:

```
cd .repo/repo/
git pull
cd ../../
.repo/repo/repo sync -l
```



```
alientek@alientek:~/ATK-DLT113IS/.repo/repo$ git pull
Updating 13d6588..db111d3
Fast-forward
 project.py      |  17 ++++++++++-----
 subcmds/gc.py   | 181 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++-------
 subcmds/sync.py |   4 +++-
 3 files changed, 188 insertions(+), 14 deletions(-)
```

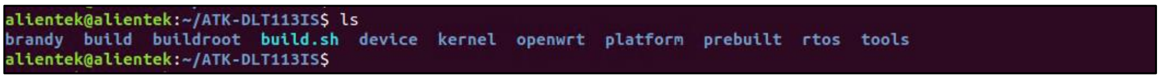
Figure 1.2-3 Update the repository



```
alientek@alientek:~/ATK-DLT113IS$ .repo/repo/repo sync
git requires authentication, but repo cannot perform interactive authentication. Check git credentials.
Fetching: 100% (107/107), done in 0.336s
Checking out files: 100% (38/38), done.-2.0/spl-pubChecking out files:  26% (26/100)
Checking out files: 100% (100/100), done.ina-ng/buildChecking out files:  71% (10242/14424)
Checking out files: 100% (14424/14424), done.ig/t113_s4pChecking out files:  62% (48/77)
Checking out files: 100% (77/77), done. -2.0/u-boot-2018Checking out files:  66% (51/77)
Checking out: 17% (19/107) lichee/brandy-2.0/toolsChecking out files: 100% (2781/2781), done.
Checking out files: 100% (2964/2964), done.a-ng/package/allwinner/tina_multimedia_demoChecking out files:  11% (41/364)
Checking out files: 100% (9/9), done.    ina-ng/package/allwinner/healthdChecking out files: 100% (9/9)
Checking out files: 100% (41/41), done.  ina-ng/package/allwinner/swupdateChecking out files:  65% (27/41)
Checking out files: 100% (44/44), done.  ina-ng/package/allwinner/nativepowerChecking out files:  75% (33/44)
Checking out files: 100% (1550/1550), done.a-ng/package/allwinner/testapkChecking out files:  21% (326/1550)
Checking out files: 100% (14/14), done.  ina-ng/package/allwinner/wifimanagerChecking out files:  50% (24/48)
Checking out files: 100% (7/7), done.)   ina-ng/package/allwinner/camerademoChecking out files:  35% (25489/72825)
Checking out files: 100% (955/955), done.ina-ng/package/thirdparty/directfbChecking out files:  35% (26035/72825)
Checking out files: 100% (48/48), done.  ina-ng/package/thirdparty/gui/lvgl-6Checking out files:  93% (45/48)
Checking out files: 100% (447/447), done.
Checking out files: 100% (220/220), done.ina-ng/package/thirdparty/bluez-alsaChecking out files:  57% (2435/4271)
Checking out files: 100% (4271/4271), done.
Checking out files: 100% (206/206), done.ina-ng/package/feeds_packagesChecking out files:  67% (40/59)
Checking out files: 100% (59/59), done./tina-ng/target/t113Checking out files:  96% (57/59)
Checking out files: 100% (274/274), done.external/libcedarseChecking out files:   8% (985/11392)
Checking out files: 100% (812/812), done.external/healthdChecking out files:  88% (715/812)
Checking out files: 100% (1498/1498), done.ternal/btmanagerChecking out files:  58% (214/364)
Checking out files: 100% (95/95), done.  external/wireless_commonChecking out files:  60% (219/364)
Checking out files: 100% (60/60), done.x/external/wifimanagerChecking out files:  51% (37664/72825)
Checking out files: 100% (1383/1383), done.ternal/wireless_firmwareChecking out files:  53% (38598/72825)
Checking out files: 100% (6/6), done.    external/littlevgl-8Checking out files:   9% (284/2934)
Checking out files: 100% (2934/2934), done.ice/chips/t113_i_c906Checking out files:  60% (44304/72825)
Checking out files: 100% (15/15), done.
Checking out files: 100% (598/598), done.evice/chips/t113_s4_c906Checking out files:  64% (46608/72825)
Checking out files: 100% (364/364), done.sChecking out files:  78% (285/364)
Checking out files: 100% (8244/8244), done.a-ng/dlChecking out files:  76% (6266/8244)
Checking out files: 100% (1830/1830), done.ponentsChecking out files:  76% (55522/72825)
Checking out files: 100% (95/95), done.  tos-halChecking out files:  78% (56804/72825)
Checking out files: 100% (72825/72825), done.-toolsChecking out files:  84% (61890/72825)
Checking out files: 100% (11392/11392), done.
Checking out files: 100% (6/6), done.nux-5.4Checking out files:  98% (848/860)
Checking out files: 100% (860/860), done.tina-ng/prebuilt/kernel-built/armChecking out files:  99% (856/860)
Checking out: 97% (104/107) lichee/toolsChecking out files: 100% (1771/1771), done.
Checking out files: 100% (5/5), done.)   tina-rtChecking out files:  80% (4/5)
Checking out files: 100% (15834/15834), done.ernal/qtChecking out files:  79% (12566/15834)
Checking out: 100% (107/107), done in 3m0.831s
repo sync has finished successfully.
alientek@alientek:~/ATK-DLT113IS$
```

Figure 1.2-4 View the source code information diagram

After the checkout is completed, we use the "ls" command to compare with the diagram in the figure to check if the folders are consistent.



```
alientek@alientek:~/ATK-DLT113IS$ ls
brandy  build  buildroot  build.sh  device  kernel  openwrt  platform  prebuilt  rtos  tools
alientek@alientek:~/ATK-DLT113IS$
```

Figure 1.2-5 The directory of the source code after being checked out

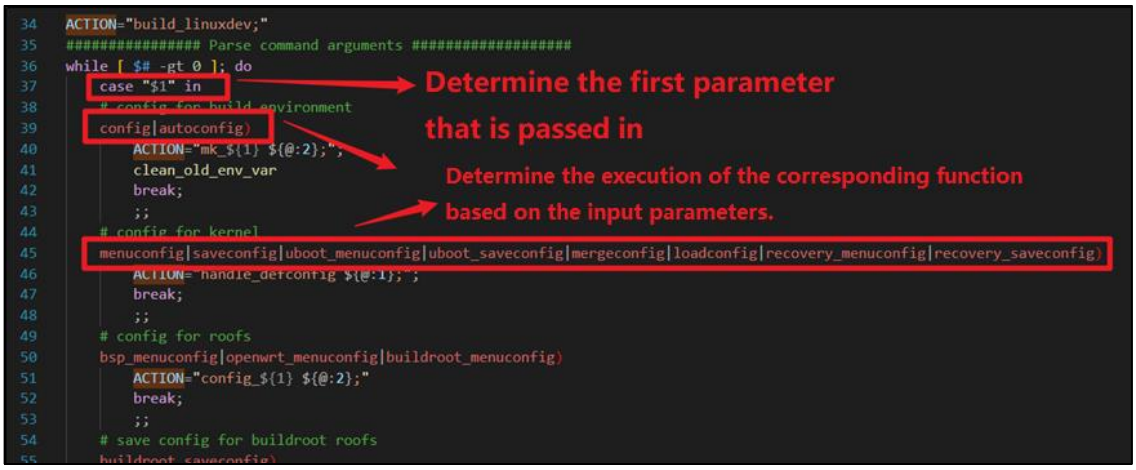
Next, let's explain the function of each file or folder in Figure 1.2.5:

- brandy: Start the relevant source code files (SPL and UBoot)
- build: Compile the relevant scripts
- buildroot: The file system of buildroot
- build.sh: The script for compilation
- device: Board-level configuration file
- kernel: Kernel source code
- openwrt: The official openwrt of Allwinner (Tina system)
- platform: Configuration or commands related to the file system (such as Allwinner's multimedia library, display library)
- prebuilt: Folder related to cross-compilers
- rtos: Directory of rtos
- tools: Configuration and compilation tools

1.2.1 Explanation of the "build" directory

There is a "build.sh" script in the source code directory. By using "ls -l", it can be seen that this file is actually a symbolic link script (build/top_build.sh). Here, I will briefly explain a few important scripts (mkcmd.sh, mkcommon.sh, pack).

- mkcommon.sh: This script is responsible for parsing the incoming parameters. For example, running "./build.sh config" where "config" represents the parameter. By examining build.sh, it can be seen that it calls the mkcommon.sh script to parse the first parameter, and it can also parse multiple parameters. The content of the mkcommon.sh script is as shown in the following figure:



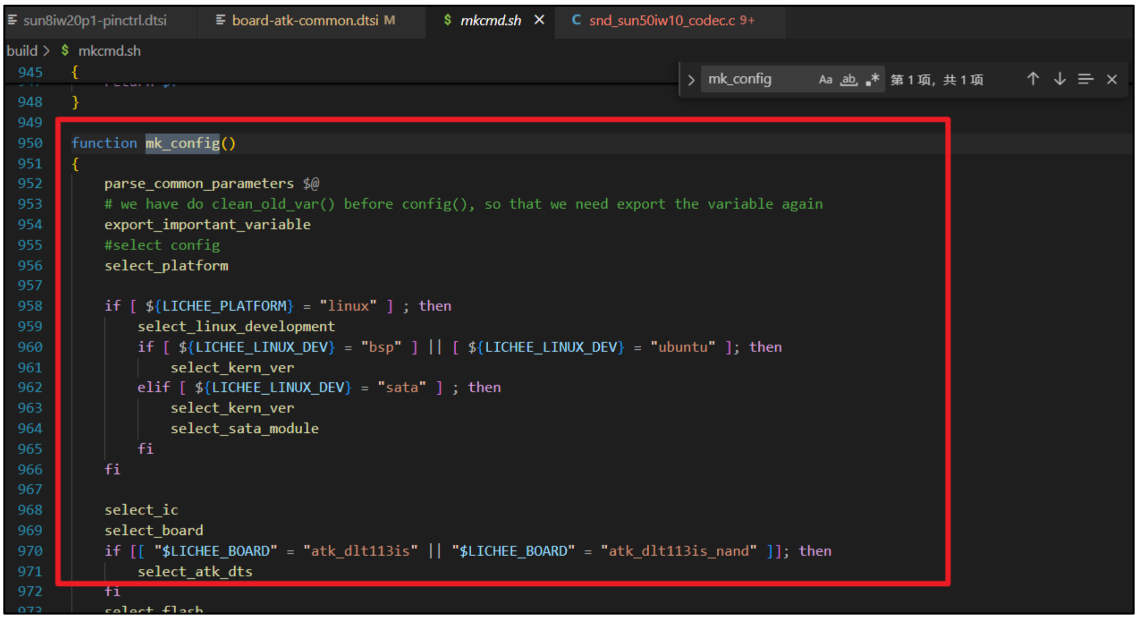
```
ACTION="build_linuxdev;"
################ Parse command arguments ###################
while [ $# -gt 0 ]; do
    case "$1" in
    # config for build environment
    config|autoconfig)
        ACTION="mk_${1} ${@:2};";
        clean_old_env_var
        break;
        ;;
    # config for kernel
    menuconfig|saveconfig|uboot_menuconfig|uboot_saveconfig|mergeconfig|loadconfig|recovery_menuconfig|recovery_saveconfig)
        ACTION="handle_defconfig ${@:1};";
        break;
        ;;
    # config for roofs
    bsp_menuconfig|openwrt_menuconfig|buildroot_menuconfig)
        ACTION="config_${1} ${@:2};"
        break;
        ;;
    # save config for buildroot roofs
    buildroot_saveconfig)
```

Determine the first parameter that is passed in

Determine the execution of the corresponding function based on the input parameters.

Figure 1.2-6 Content illustration of the mkcommon.sh script

- mkcmd.sh: This script is responsible for running the corresponding function based on the parameters parsed by mkcommon.sh. For example: Running "./build.sh config", after being parsed by mkcommon.sh, the ACTION parameter is "mk_config". Open the mkcmd.sh file to find this function, as shown in the following figure:



```
{
}

function mk_config()
{
    parse_common_parameters $@
    # we have do clean_old_var() before config(), so that we need export the variable again
    export_important_variable
    #select config
    select_platform

    if [ ${LICHEE_PLATFORM} = "linux" ] ; then
        select_linux_development
        if [ ${LICHEE_LINUX_DEV} = "bsp" ] || [ ${LICHEE_LINUX_DEV} = "ubuntu" ]; then
            select_kern_ver
        elif [ ${LICHEE_LINUX_DEV} = "sata" ] ; then
            select_kern_ver
            select_sata_module
        fi
    fi

    select_ic
    select_board
    if [[ "$LICHEE_BOARD" = "atk_dlt113is" || "$LICHEE_BOARD" = "atk_dlt113is_nand" ]]; then
        select_atk_dts
    fi
    select_flash
```

Figure 1.2-7 Content of the mkcmd.sh script

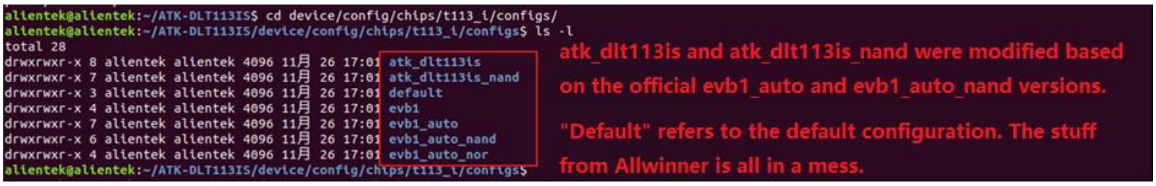
pack: This file is also a script. Its main function is to perform system packaging after the compilation is completed.

The basic introduction of the commonly used scripts here is complete. The mkcommon.sh script is responsible for parsing the parameters and running different operations based on different parameters. For example: Running "./build.sh menuconfig" as students who have learned about 6u know, there are two methods for configuring the kernel driver. The most commonly used one is to run "make menuconfig" in the source code directory of the kernel for configuration. "./build.sh menuconfig" is the process of configuring the kernel driver.

1.2.2 Explanation of the "device" Directory

The "device" directory holds a crucial position in the development process of the Allwinner SDK, as it contains the configuration information for the Allwinner T113 series. Given that the SDK is designed for a series of chips, when using the SDK, we must select the corresponding configuration based on the chip we are using. Taking the ALIENTEK ATK-DLT113IS development board as an example, it uses the T113-I chip. Therefore, we need to find the specific configuration for the T113-I chip from the "device" directory to ensure that the development board can run correctly. Navigate to the following path in the source code directory of the SDK: "device/config/chips/t113_i/configs/". Run the "ls" command to view the folders and execute the code as follows:

```
cd device/config/chips/t113_i/configs/
ls -l
```



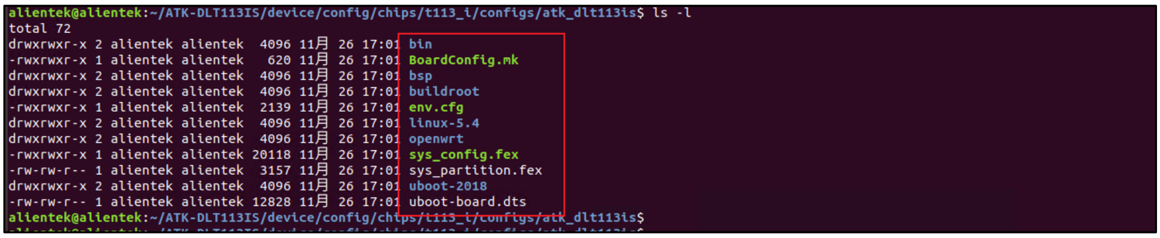
```
alientek@alientek:~/ATK-DLT113IS$ cd device/config/chips/t113_i/configs/
alientek@alientek:~/ATK-DLT113IS/device/config/chips/t113_i/configs$ ls -l
total 28
drwxrwxr-x 8 alientek alientek 4096 11月 26 17:01 atk_dlt113is
drwxrwxr-x 7 alientek alientek 4096 11月 26 17:01 atk_dlt113is_nand
drwxrwxr-x 3 alientek alientek 4096 11月 26 17:01 default
drwxrwxr-x 4 alientek alientek 4096 11月 26 17:01 evb1
drwxrwxr-x 7 alientek alientek 4096 11月 26 17:01 evb1_auto
drwxrwxr-x 6 alientek alientek 4096 11月 26 17:01 evb1_auto_nand
drwxrwxr-x 4 alientek alientek 4096 11月 26 17:01 evb1_auto_nor
alientek@alientek:~/ATK-DLT113IS/device/config/chips/t113_i/configs$
```

atk_dlt113is and atk_dlt113is_nand were modified based on the official evb1_auto and evb1_auto_nand versions.

"Default" refers to the default configuration. The stuff from Allwinner is all in a mess.

Figure 1.2-8 Configuration folder diagram of T113-I

There are two storage methods for ATK-DLT113IS: EMMC and SPI NAND. Developers purchasing the EMMC version should use the atk_dlt113is configuration, while those with the SPI NAND version should use atk_dlt113is_nand. Here, I will give a brief explanation using the atk_dlt113is folder (the atk_dlt113is and atk_dlt113is_nand folders are quite similar and will not be explained one by one). Go to this "atk_dlt113is" directory:



```
alientek@alientek:~/ATK-DLT113IS/device/config/chips/t113_i/configs/atk_dlt113is$ ls -l
total 72
drwxrwxr-x 2 alientek alientek  4096 11月 26 17:01 bin
-rwxrwxr-x 1 alientek alientek   620 11月 26 17:01 BoardConfig.mk
drwxrwxr-x 2 alientek alientek  4096 11月 26 17:01 bsp
drwxrwxr-x 2 alientek alientek  4096 11月 26 17:01 buildroot
-rwxrwxr-x 1 alientek alientek  2139 11月 26 17:01 env.cfg
drwxrwxr-x 2 alientek alientek  4096 11月 26 17:01 linux-5.4
drwxrwxr-x 2 alientek alientek  4096 11月 26 17:01 openwrt
-rwxrwxr-x 1 alientek alientek 20118 11月 26 17:01 sys_config.fex
-rw-rw-r-- 1 alientek alientek  3157 11月 26 17:01 sys_partition.fex
drwxrwxr-x 2 alientek alientek  4096 11月 26 17:01 uboot-2018
-rw-rw-r-- 1 alientek alientek 12828 11月 26 17:01 uboot-board.dts
alientek@alientek:~/ATK-DLT113IS/device/config/chips/t113_i/configs/atk_dlt113is$
```

Figure 1.2-9 Content diagram of the "atk_dlt113is" folder

In Figure 1.2.9, there is a lot of valuable information. The explanations for each file:

- bin: It stores the RTOS firmware of the C906 core.
- BoardConfig.mk: The board-level configuration file. There are several of these files, and the one used will be determined based on your configuration. For example: If the configuration

file system is buildroot, the build.sh script will use "buildroot/BoardConfig.mk", so it is determined according to the actual situation.

- buildroot: The configuration of buildroot.
- bsp: The configuration of bsp.
- env.cfg: This is related to Uboot and is an environment variable file. This file, like BoardConfig.mk, will have different env.cfg files used depending on the different configurations.
- linux-5.4: Here are the configuration files and kernel device trees of the kernel.
- openwrt: The configuration of openwrt.
- sys_config.fex: This is a feature of ZTE. The most basic configuration file, used to configure the peripheral files related to BOOT0 startup.
- sys_partition.fex: This is a configuration file related to ZTE's image partition, and like env.cfg, there are multiple files, and the one used will be determined based on the actual configuration.
- uboot-2018: This file was added by the author. It stores all the device trees of Uboot.
- uboot-board.dts: The official Uboot device tree of ZTE

Note: The three files are: BoardConfig.mk, env.cfg and sys_partition.fex. They are all stored in the "atk_dlt113is" folder and multiple files are placed in each of them. The specific file to be used will be determined based on different configurations. By selecting the buildroot file system, these three files in the buildroot directory will be called.

1. buildroot folder

This folder contains the relevant configurations for buildroot. When the selected buildroot is used as the file system, the files in this folder will be called. The content of the BoardConfig.mk file in the buildroot directory is as shown in the following example code:

Example code 1.2.1 Content of BoardConfig.mk file

```
1    LICHEE_CHIP:=sun8iw20p1
2    LICHEE_PRODUCT:=atk_dlt113is
3    LICHEE_BOARD:=atk_dlt113is
4    LICHEE_FLASH:=
5    LICHEE_ARCH:=arm
6    LICHEE_KERN_DEFCONF:=alientek_t113_kernel_defconfig
7    #LICHEE_KERN_DEFCONF_RECOVERY:=sun8iw20p1smp_t113_recovery_defconfig
8    LICHEE_BUILDING_SYSTEM:=buildroot
9    LICHEE_ATK_KERN_COMMON:=board-atk-common.dtsi
10   LICHEE_BR_VER:=201902
11   LICHEE_BRANDY_UBOOT_VER:=2018
12   LICHEE_BR_DEFCONF:=alientek_t113_br_defconfig
13   LICHEE_COMPILER_TAR=arm/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-
gnueabi.tar.xz
14   LICHEE_BRANDY_DEFCONF:=alientek_t113_uboot_defconfig
15   LICHEE_REDUNDANT_ENV_SIZE:=0x20000
16   LICHEE_RTOS_PROJECT_NAME:=t113_i_c906_evb1_auto
```

Line 1: The IP of the chip is "sun8iw20p1".

Line 6: The configuration file for the kernel is set to "alientek_t113_kernel_defconfig", which is located in the "linux-5.4" directory.

Line 7: Commented out; do not pay attention to it.

Line 8: Set buildroot as the file system.

Line 9: This variable was added by me; it sets the general configuration peripherals for the ALIENTEK.

Line 10: Set the version of buildroot to 201902 (used by Allwinner).

Line 11：Set the version of u-boot to 2018.

Line 12：Set the configuration file of buildroot to "alientek_t113_br_defconfig".

Line 13：Set the compilation version of the kernel to "gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi".

Line 14：Set the configuration file of u-boot to "alientek_t113_uboot_defconfig".

Line 15：Here， the size of the environment variables is defined as 0x20000.

Line 16：The configuration file of the RTOS is "t113_i_c906_evb1_auto".

In the env.cfg file located in the buildroot directory, the content is as shown in the following example code:

Example code 1.2.2 Content of the env.cfg file

```
#kernel command arguments
earlycon=uart8250,mmio32,0x02500000
initcall_debug=0
console=ttyAS0,115200
nand_root=ubi0_5
mmc_root=/dev/mmcblk0p5
mtd_name=sys
rootfstype=ubifs,rw
init=/init
loglevel=8
cma=16M
mac=
wifi_mac=
bt_mac=
specialstr=
keybox_list=widevine,ec_key,ec_cert1,ec_cert2,ec_cert3,rsa_key,rsa_cert1,rsa_cert2,rsa_cert3
dsp0_partition=dsp0
#set kernel cmdline if boot.img or recovery.img has no cmdline we will use this
setargs_nand=setenv bootargs ubi.mtd=${mtd_name} earlycon=${earlycon} clk_ignore_unused initcall_debug=${initcall_debug} console=${console} loglevel=${loglevel} root=${nand_root} rootfstype=${rootfstype} rootwait init=${init} partitions=${partitions}
```

```
cma=${cma} snum=${snum} mac_addr=${mac} wifi_mac=${wifi_mac} bt_mac=${bt_mac} specialstr=${specialstr} gpt=1
20  setargs_nand_ubi=setenv bootargs ubi.mtd=${mtd_name} earlycon=${earlycon} clk_ignore_unused initcall_debug=${initcall_debug} console=${console} loglevel=${loglevel} root=${nand_root} rootfstype=${rootfstype} rootwait init=${init} partitions=${partitions} cma=${cma} snum=${snum} mac_addr=${mac} wifi_mac=${wifi_mac} bt_mac=${bt_mac} specialstr=${specialstr} gpt=1
21  setargs_mmc=setenv  bootargs earlycon=${earlycon} clk_ignore_unused initcall_debug=${initcall_debug} console=${console} loglevel=${loglevel} root=${mmc_root}  rootwait init=${init} partitions=${partitions} cma=${cma} snum=${snum} mac_addr=${mac} wifi_mac=${wifi_mac} bt_mac=${bt_mac} specialstr=${specialstr} gpt=1
22  #nand command syntax: sunxi_flash read address partition_name read_bytes
23  #0x4007f800 = 0x40080000(kernel entry) - 0x800(boot.img header 2k)
24  boot_dsp0=sunxi_flash read 43000000 ${dsp0_partition};bootr 43000000 0 0
25  boot_normal=sunxi_flash read 43000000 boot;bootm 43000000
26  boot_recovery=sunxi_flash read 43000000 recovery;bootm 43000000
27  boot_fastboot=fastboot
28
29  #uboot system env config
30  bootdelay=0
31  #default bootcmd, will change at runtime according to key press
32  #default nand boot
33  bootcmd=run setargs_mmc boot_normal
```

Line 2: The internal initialization serial port uses the kernel's common 8250 serial port controller.

Line 4: The debugging serial port of ATK-DLT113IS is uart0, so the debugging serial port is ttyAS0, with a baud rate of 115200. If using the full-vision controller, the serial port is ttyS0.

Line 5: The file system partition of NAND is ubi0_5.

Line 6: The file system of eMMC, the mounted partition is "/dev/mmcblk0p5".

Line 8: The type of the file system is ubifs, allowing read and write.

Line 30: The delay of bootdelay is 0.

Line 33: After the bootdelay time is over, the bootcmd is executed.

The content of the "sys_partition.fex" file in the "buildroot" directory is as follows (as shown in the example code):

Example code 1.2.3 Content of the sys_partition.fex file

```
15 ;***************************************************************************************************
16 ;                                      partitioned allocation
17 ;
```

```
;
; partition Definition example:
;    [partition]             ;  // It indicates a partition.
;    name       = USERFS2     ; // partition name
;    size       = 16384       ; // Partition size Unit: Sector. The maximum number of partitions is 2^31 * 512 = 2T
;    downloadfile = "123.fex"   ; // The path and name of the downloaded file can be in a relative format. "Relative" here means relative to the partition where the "image.cfg" file is located. Alternatively, an absolute path can also be used.
;    keydata     = 1           ; // Private data partitioning ensures that re-production of data will not result in loss.
;    encrypt     = 1           ; // Using an encrypted method for burning will provide data encryption, but it will result in a reduction in burning speed.
;    user_type   = ?           ; // Personal use
;    verify      = 1           ; // After the mass production is completed, verify whether it is correct.
;
; Note: 1. The name must be unique and duplicate names are not allowed.
;       2、name maximum 12 characters
;       3、size = 0, Create an empty partition with no size.
;       4、align to logical block size(504 sectors), leb size = 2*(1 nand phy block size - 1 phy page size)

;***************************************************************************************************
[partition_start]

[partition]
    name         = boot-resource
    size         = 34438
    downloadfile = "boot-resource.fex"
    user_type    = 0x8000


[partition]
    name         = env
    size         = 2048
    downloadfile = "env.fex"
    user_type    = 0x8000

```

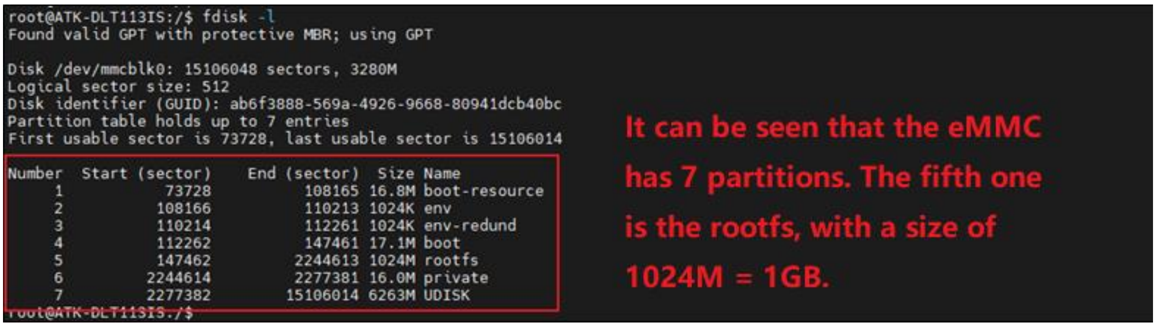
```
49 [partition]
50     name        = env-redund
51     size        = 2048
52     downloadfile = "env.fex"
53     user_type    = 0x8000
54
55 [partition]
56     name        = boot
57     size        = 35200
58     downloadfile = "boot.fex"
59     user_type    = 0x8000
60
61 [partition]
62     name        = rootfs
63     size        = 2097152
64     downloadfile = "rootfs.fex"
65     user_type    = 0x8000
66
67 ;[partition]
68 ;    name        = dsp0
69 ;    size        = 2048
70 ;    downloadfile = "amp_dsp0.fex"
71 ;    user_type    = 0x8000
72
73 [partition]
74     name        = private
75     size        = 32768
76     ro          = 0
77     user_type    = 0x8000
78
79 [partition]
80     name        = UDISK
81     user_type    = 0x8100
```

Lines 19 to 32 provide an explanation of how this file is used and how to customize your own partition. When the system is packaged, this file is called to create 7 partitions: boot-resource, env, env-redund, boot, rootfs, private, and UDISK (the dsp0 partition is commented). downloadfile represents the path and name of the file to be downloaded. size indicates the size of the partition, measured in 512 bytes (note that it is lowercase 'b'), so the size of the rootfs in the file system is:

```
2097152 x 512b = 1073741824b
```

By unit conversion, we know that 1GB. To start the development board and run the following command to view the partitions and their sizes:

```
fdisk -l
```



```
root@ATK-DLT113IS:/$ fdisk -l
Found valid GPT with protective MBR; using GPT

Disk /dev/mmcblk0: 15106048 sectors, 3280M
Logical sector size: 512
Disk identifier (GUID): ab6f3888-569a-4926-9668-80941dcb40bc
Partition table holds up to 7 entries
First usable sector is 73728, last usable sector is 15106014

Number  Start (sector)    End (sector)  Size Name
     1           73728          108165 16.8M boot-resource
     2          108166          110213 1024K env
     3          110214          112261 1024K env-redund
     4          112262          147461 17.1M boot
     5          147462         2244613 1024M rootfs
     6         2244614         2277381 16.0M private
     7         2277382        15106014 6263M UDISK
root@ATK-DLT113IS:/$
```

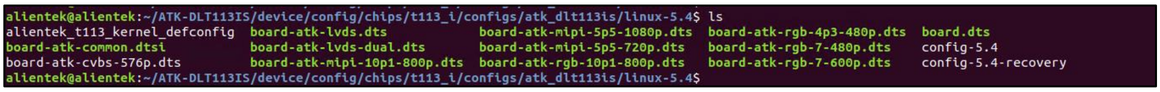
It can be seen that the eMMC has 7 partitions. The fifth one is the rootfs, with a size of 1024M = 1GB.

Figure 1.2-10 View partition size chart

Set user_type to private use directly and configure it as 0x8000.

2. linux-5.4 folder

This folder stores kernel-related files. Go to it and view the results by using "ls". The result is shown in the following chart:



```
alientek@alientek:~/ATK-DLT113IS/device/config/chips/t113_i/configs/atk_dlt113is/linux-5.4$ ls
alientek_t113_kernel_defconfig  board-atk-lvds.dts            board-atk-mipi-5p5-1080p.dts  board-atk-rgb-4p3-480p.dts  board.dts
board-atk-common.dtsi           board-atk-lvds-dual.dts       board-atk-mipi-5p5-720p.dts   board-atk-rgb-7-480p.dts   config-5.4
board-atk-cvbs-576p.dts         board-atk-mipi-10p1-800p.dts  board-atk-rgb-10p1-800p.dts   board-atk-rgb-7-600p.dts   config-5.4-recovery
alientek@alientek:~/ATK-DLT113IS/device/config/chips/t113_i/configs/atk_dlt113is/linux-5.4$
```

Figure 1.2-11 The "linux-5.4" folder

"alientek_t113_kernel_defconfig"; represents the kernel configuration file for the ATK-DLT113IS development board.

"board-atk-common.dtsi": represents the default board-level configuration device tree for the ATK-DLT113IS development board.

The subsequent "board-atk-xxxx.dts" device tree files are adapted to different display interfaces and customized for different hardware configurations. The ATK-DLT113IS reserves all display interfaces, which results in many screen configuration device trees. For example, the 7-inch 1024x600 RGB screen you purchased uses "board-atk-rgb-7-600p.dts".

"board-atk-cvbs-576p.dts": uses the official CVBS interface for display output. We do not sell the corresponding hardware. Please purchase it from Taobao yourself.

"board-atk-lvds.dts": adapted to the ALIENTEK LVDS screen.

"board-atk-lvds-dual.dts": 8-lane LVDS screen configuration, tested during development.

"board-atk-mipi-10p1-800p.dts": adapted to the MIPI 10-inch screen of ALIENTEK.

"board-atk-mipi-5p5-1080p.dts": adapted to the MIPI 5.5-inch 1080p screen of ALIENTEK.

"board-atk-mipi-5p5-720p.dts": adapted to the MIPI 5.5-inch 720p screen of ALIENTEK.

"board-atk-rgb-10p1-800p.dts": adapted to the RGB 10-inch 800p screen of ALIENTEK.

"board-atk-rgb-4p3-480p.dts": adapted to the RGB 4.3-inch 480p screen of ALIENTEK.

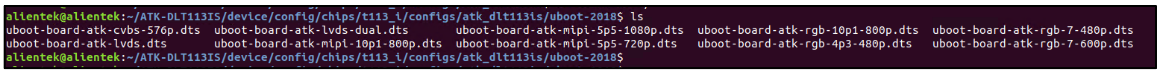
"board-atk-rgb-7-480p.dts": adapted to the RGB 7-inch 480p screen of ALIENTEK.

board-atk-rgb-7-600p.dts: This is adapted for the RGB 7-inch 600p screen of the ALIENTEK.

board.dts, config-5.4 and config-5.4-recovery are the original files of Allwinner.

3. uboot-2018 folder

This folder is customized by me. There's nothing much to say about it. The uboot device trees supporting different screens are as shown in the following figure:



```
alientek@alientek:~/ATK-DLT113IS/device/config/chips/t113_i/configs/atk_dlt113is/uboot-2018$ ls
uboot-board-atk-cvbs-576p.dts  uboot-board-atk-lvds-dual.dts       uboot-board-atk-mipi-5p5-1080p.dts  uboot-board-atk-rgb-10p1-800p.dts  uboot-board-atk-rgb-7-480p.dts
uboot-board-atk-lvds.dts       uboot-board-atk-mipi-10p1-800p.dts  uboot-board-atk-mipi-5p5-720p.dts   uboot-board-atk-rgb-4p3-480p.dts   uboot-board-atk-rgb-7-600p.dts
alientek@alientek:~/ATK-DLT113IS/device/config/chips/t113_i/configs/atk_dlt113is/uboot-2018$
```

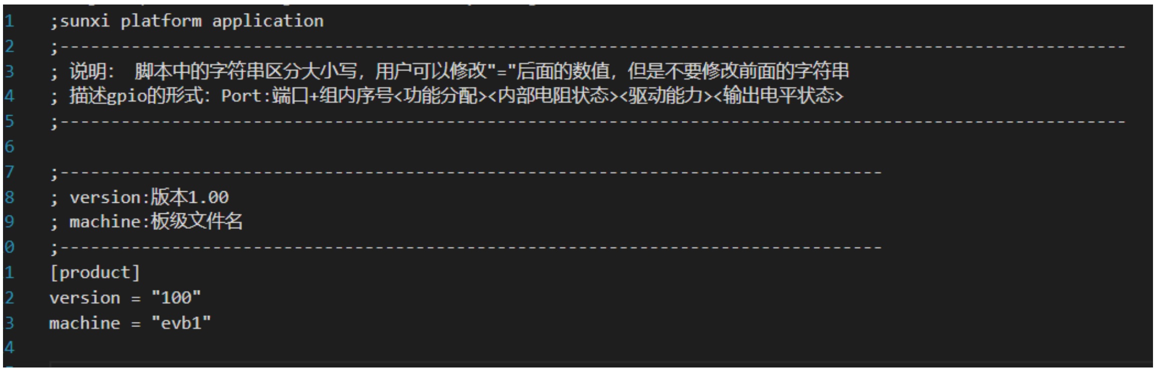
Figure 1.2-12 uboot-2018 folder diagram

4. openwrt folder

openwrt has only one file named BoardConfig.mk. This file is basically the same as the BoardConfig.mk file in the buildroot directory. When we choose to configure, we call the "openwrt/BoardConfig.mk" file.

5. sys_config.fex file

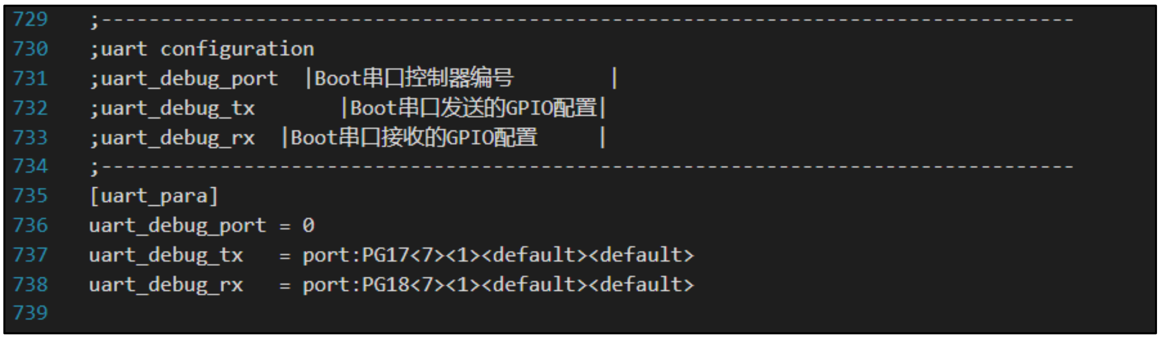
This file is the hardware peripheral description of the ZTE Zhonghua. View the content of this file as shown in the following diagram:



```
1    ;sunxi platform application
2    ;-------------------------------------------------------------------------------------------------
3    ; 说明： 脚本中的字符串区分大小写，用户可以修改"="后面的数值，但是不要修改前面的字符串
4    ; 描述gpio的形式：Port:端口+组内序号<功能分配><内部电阻状态><驱动能力><输出电平状态>
5    ;-------------------------------------------------------------------------------------------------
6
7    ;--------------------------------------------------------------------------
8    ; version:版本1.00
9    ; machine:板级文件名
0    ;--------------------------------------------------------------------------
1    [product]
2    version = "100"
3    machine = "evb1"
4
```

Figure 1.2-13 Content of the sys_config file

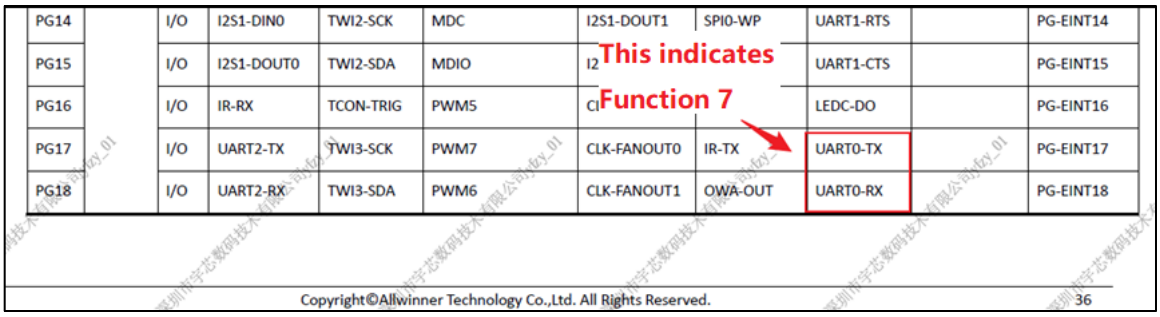
The comment on the 4th line: This describes how to define the function of an IO. Here, the author takes the serial port as an example. As shown in the figure below:



```
729   ;--------------------------------------------------------------------------------
730   ;uart configuration
731   ;uart_debug_port  |Boot串口控制器编号        |
732   ;uart_debug_tx       |Boot串口发送的GPIO配置|
733   ;uart_debug_rx  |Boot串口接收的GPIO配置     |
734   ;--------------------------------------------------------------------------------
735   [uart_para]
736   uart_debug_port = 0
737   uart_debug_tx   = port:PG17<7><1><default><default>
738   uart_debug_rx   = port:PG18<7><1><default><default>
739
```

Figure 1.2-14 Serial Port Description Diagram

From the comments, it can be seen that at line 736, uart0 is used as boot0 for starting and the printing information is shown. At lines 737 and 738, it describes the use of PG17 and PG18 as TX and RX respectively. In Figure 1.2.13, PG represents the port, and 17 and 18 indicate the sequence number within the group. The first < > indicates the multiplexing of functions. Both PG17 and PG18 are successfully multiplexed to function 7. This can be known from the "Datasheet". In the data disk folder: Development Board CD-ROM A Disk - Basic Materials →7_Allwinner_reference → T113-i_Datasheet_V1.8.pdf. Open this PDF file and jump to page 47. As shown in the following figure:



PG14		I/O	I2S1-DIN0	TWI2-SCK	MDC	I2S1-DOUT1	SPI0-WP	UART1-RTS		PG-EINT14
PG15		I/O	I2S1-DOUT0	TWI2-SDA	MDIO	I2		UART1-CTS		PG-EINT15
PG16		I/O	IR-RX	TCON-TRIG	PWM5	Cl		LEDC-DO		PG-EINT16
PG17		I/O	UART2-TX	TWI3-SCK	PWM7	CLK-FANOUT0	IR-TX	UART0-TX		PG-EINT17
PG18		I/O	UART2-RX	TWI3-SDA	PWM6	CLK-FANOUT1	OWA-OUT	UART0-RX		PG-EINT18

This indicates Function 7

Copyright©Allwinner Technology Co.,Ltd. All Rights Reserved. 36

Figure 1.2-15 Function description of the datasheet

By viewing the "Datasheet", it can be known that the function 7 of PG17 and PG18 is the TX and RX of the serial port. If you need to modify the corresponding function reassignment, please refer to the "Datasheet" first. The remaining <> symbols in the following part, the author couldn't find the official materials. Please search on the internet by yourselves.

1.3 Explanation of Parameters in build.sh

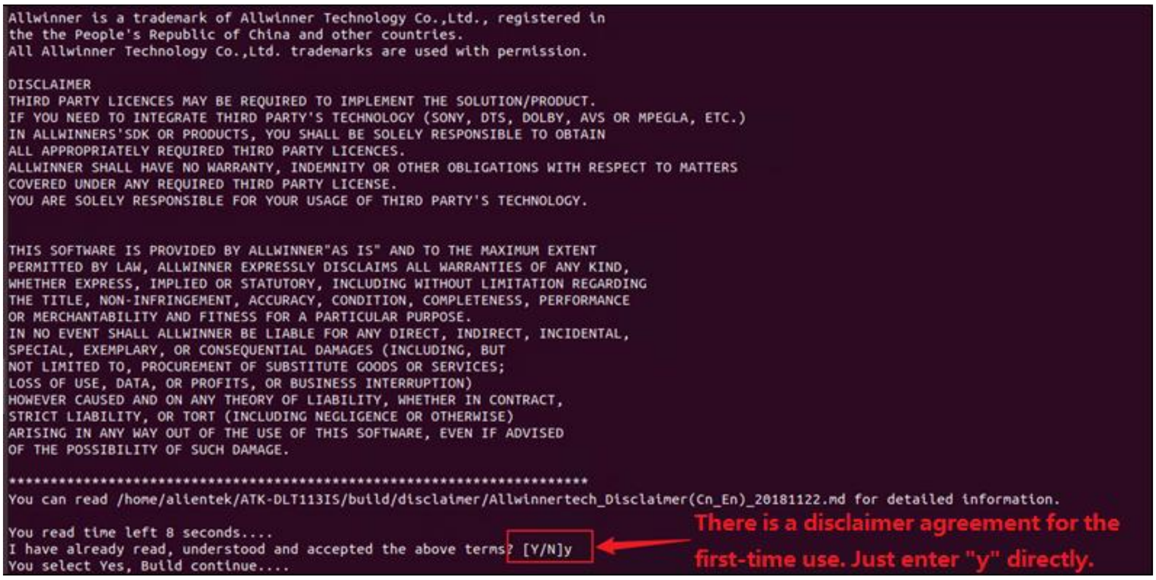
The author is explaining based on the emmc version of ATK-DLT113IS. Therefore, when making configuration selections, all options related to emmc should be chosen. If you purchase a NAND version, please modify the configuration yourself.

The subsequent 1.3.4 operation of compilation requires full compilation as in 1.3.2.

1.3.1 Configuration

Open the terminal in the SDK source code directory and then run the following code:

```
./build.sh config
```

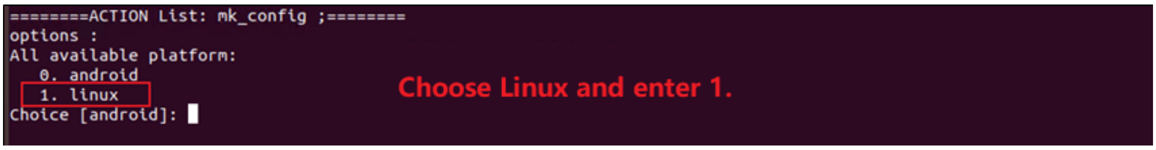


```
Allwinner is a trademark of Allwinner Technology Co.,Ltd., registered in
the the People's Republic of China and other countries.
All Allwinner Technology Co.,Ltd. trademarks are used with permission.

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 /home/alientek/ATK-DLT113IS/build/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]y
You select Yes, Build continue....
```

There is a disclaimer agreement for the first-time use. Just enter "y" directly.

Figure 1.3-1 Agree to the liability waiver agreement

When "y" is selected, the following options will pop up. Simply select "linux":

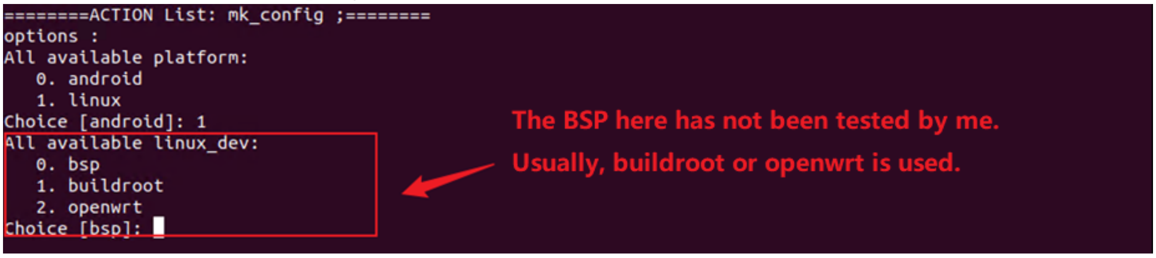


```
========ACTION List: mk_config ;========
options :
All available platform:
   0. android
   1. linux
Choice [android]: 
```

Choose Linux and enter 1.

Figure 1.3-2 Select platform

The T113 series does not have Android. There is an option for Android in the SDK, which seems to be reserved for other chips by them. Do not choose Android at all.

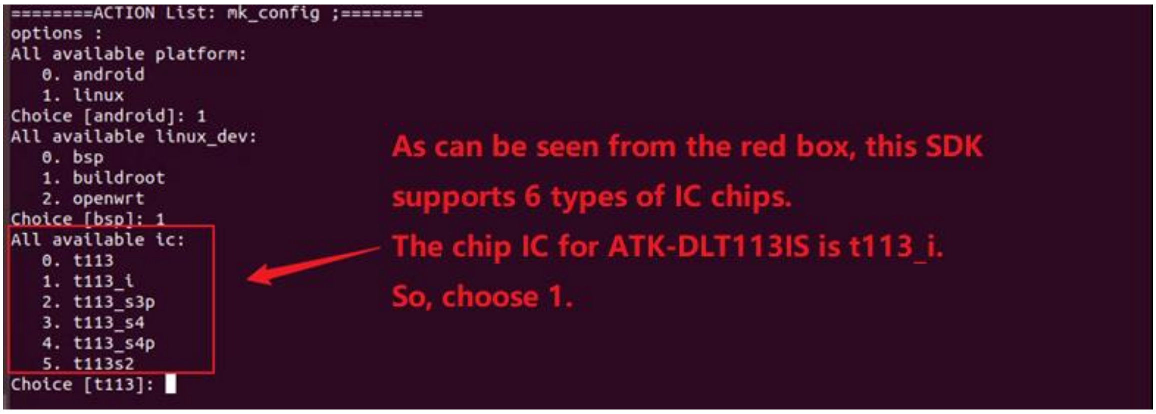


```
========ACTION List: mk_config ;========
options :
All available platform:
   0. android
   1. linux
Choice [android]: 1
All available linux_dev:
   0. bsp
   1. buildroot
   2. openwrt
Choice [bsp]: 
```

The BSP here has not been tested by me.
Usually, buildroot or openwrt is used.

Figure 1.3-3 Select "linux_dev"

Here, we choose option 1 and use buildroot. Then, a prompt to select the chip type will appear.

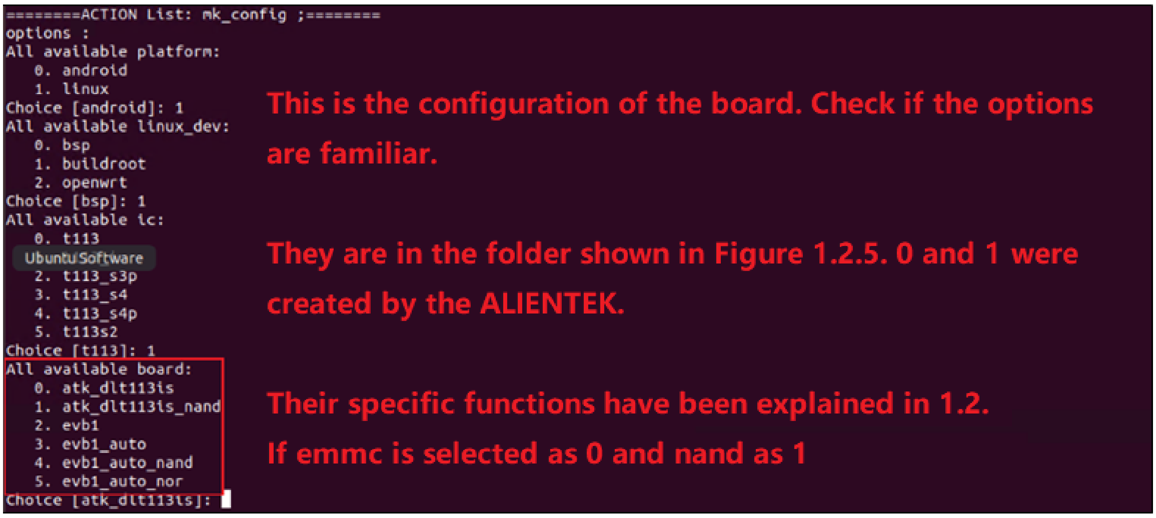


```
========ACTION List: mk_config ;========
options :
All available platform:
   0. android
   1. linux
Choice [android]: 1
All available linux_dev:
   0. bsp
   1. buildroot
   2. openwrt
Choice [bsp]: 1
All available ic:
   0. t113
   1. t113_i
   2. t113_s3p
   3. t113_s4
   4. t113_s4p
   5. t113s2
Choice [t113]: 
```

As can be seen from the red box, this SDK supports 6 types of IC chips.
The chip IC for ATK-DLT113IS is t113_i.
So, choose 1.

Figure 1.3-4 Select IC

In Figure 1.3.4, select "t113_i". Then a board-level configuration information selection window will pop up:



```
========ACTION List: mk_config ;========
options :
All available platform:
   0. android
   1. linux
Choice [android]: 1
All available linux_dev:
   0. bsp
   1. buildroot
   2. openwrt
Choice [bsp]: 1
All available ic:
   0. t113
   2. t113_s3p
   3. t113_s4
   4. t113_s4p
   5. t113s2
Choice [t113]: 1
All available board:
   0. atk_dlt113is
   1. atk_dlt113is_nand
   2. evb1
   3. evb1_auto
   4. evb1_auto_nand
   5. evb1_auto_nor
Choice [atk_dlt113is]: 
```

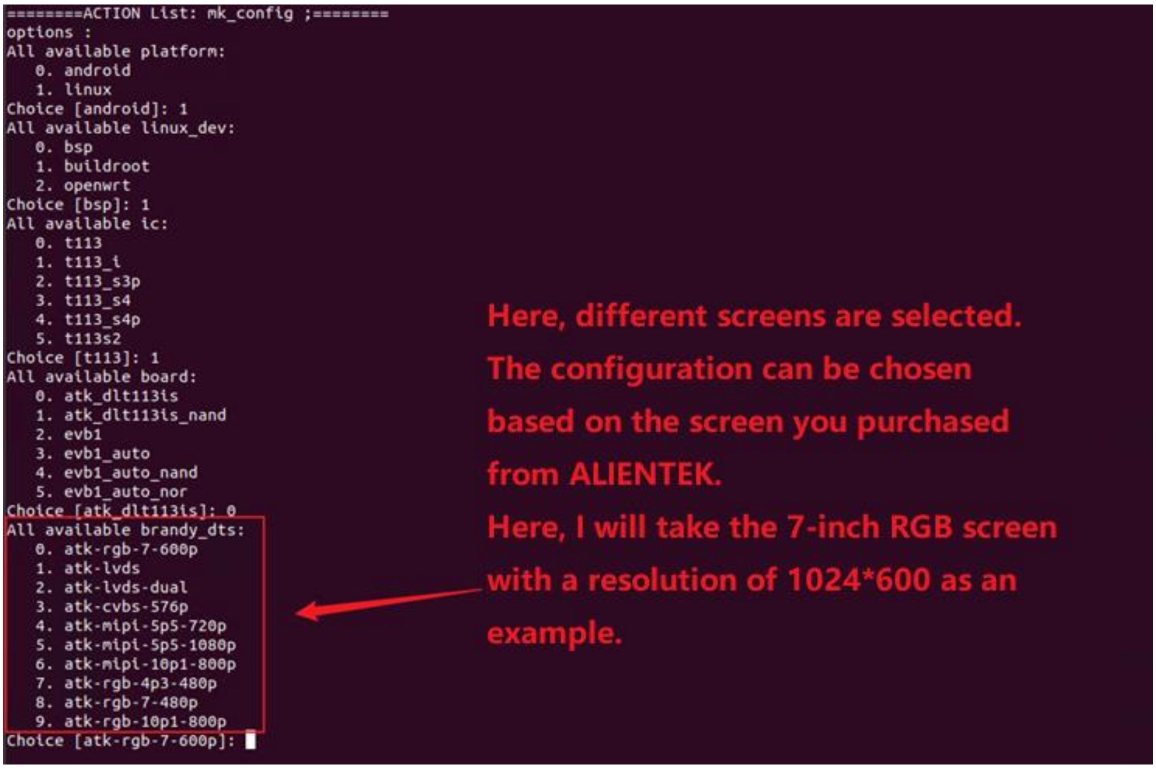
This is the configuration of the board. Check if the options are familiar.

They are in the folder shown in Figure 1.2.5. 0 and 1 were created by the ALIENTEK.

Their specific functions have been explained in 1.2.
If emmc is selected as 0 and nand as 1

Figure 1.3-5 Board selection

The author is using the eMMC version. Here, "0" is selected. Then, the screen-related configurations are chosen (no screen compatibility was performed, so each screen has its own device tree).



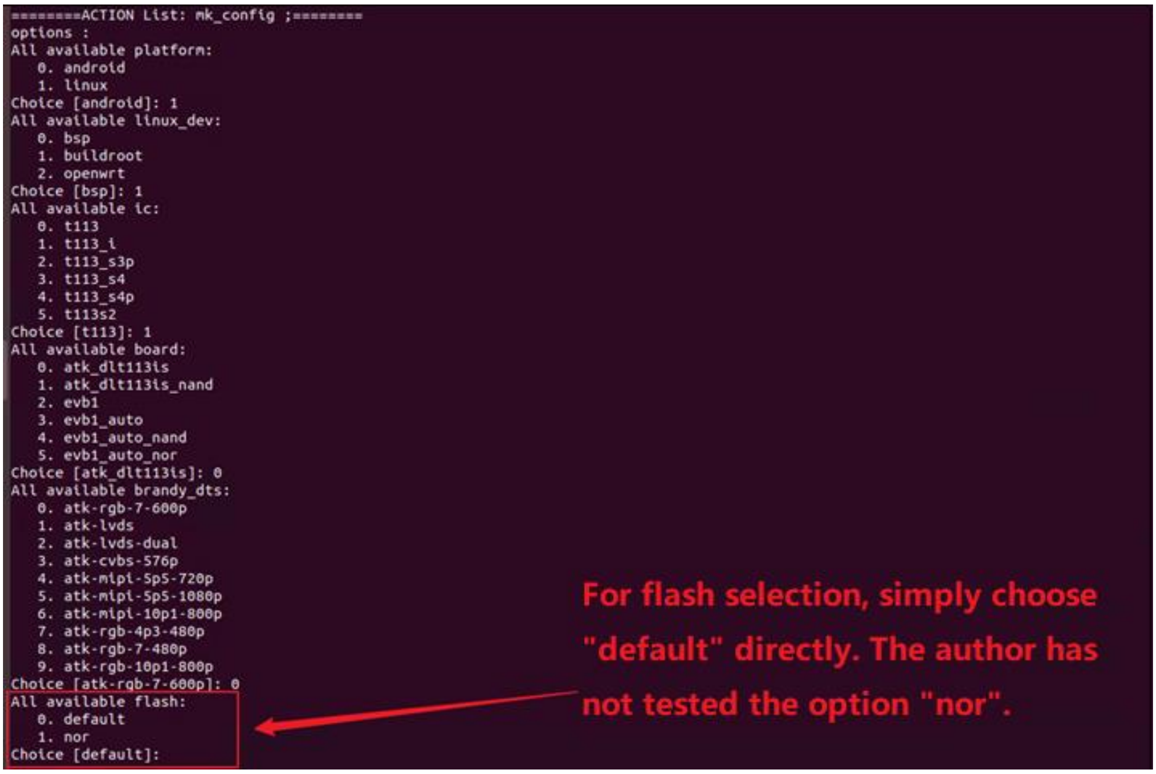
```
========ACTION List: mk_config ;========
options :
All available platform:
   0. android
   1. linux
Choice [android]: 1
All available linux_dev:
   0. bsp
   1. buildroot
   2. openwrt
Choice [bsp]: 1
All available ic:
   0. t113
   1. t113_i
   2. t113_s3p
   3. t113_s4
   4. t113_s4p
   5. t113s2
Choice [t113]: 1
All available board:
   0. atk_dlt113is
   1. atk_dlt113is_nand
   2. evb1
   3. evb1_auto
   4. evb1_auto_nand
   5. evb1_auto_nor
Choice [atk_dlt113is]: 0
All available brandy_dts:
   0. atk-rgb-7-600p
   1. atk-lvds
   2. atk-lvds-dual
   3. atk-cvbs-576p
   4. atk-mipi-5p5-720p
   5. atk-mipi-5p5-1080p
   6. atk-mipi-10p1-800p
   7. atk-rgb-4p3-480p
   8. atk-rgb-7-480p
   9. atk-rgb-10p1-800p
Choice [atk-rgb-7-600p]:
```

Here, different screens are selected. The configuration can be chosen based on the screen you purchased from ALIENTEK.

Here, I will take the 7-inch RGB screen with a resolution of 1024*600 as an example.

Figure 1.3-6 Brandy DTS selection

Here, select "0", and finally just choose "default" 0.



```
========ACTION List: mk_config ;========
options :
All available platform:
   0. android
   1. linux
Choice [android]: 1
All available linux_dev:
   0. bsp
   1. buildroot
   2. openwrt
Choice [bsp]: 1
All available ic:
   0. t113
   1. t113_i
   2. t113_s3p
   3. t113_s4
   4. t113_s4p
   5. t113s2
Choice [t113]: 1
All available board:
   0. atk_dlt113is
   1. atk_dlt113is_nand
   2. evb1
   3. evb1_auto
   4. evb1_auto_nand
   5. evb1_auto_nor
Choice [atk_dlt113is]: 0
All available brandy_dts:
   0. atk-rgb-7-600p
   1. atk-lvds
   2. atk-lvds-dual
   3. atk-cvbs-576p
   4. atk-mipi-5p5-720p
   5. atk-mipi-5p5-1080p
   6. atk-mipi-10p1-800p
   7. atk-rgb-4p3-480p
   8. atk-rgb-7-480p
   9. atk-rgb-10p1-800p
Choice [atk-rgb-7-600p]: 0
All available flash:
   0. default
   1. nor
Choice [default]:
```

For flash selection, simply choose "default" directly. The author has not tested the option "nor".

Figure 1.3-7 Flash selection

After we complete the configuration, a "out" directory will be generated. All the configuration files of Uboot, kernel, and file system will be generated with the default settings. When we modify the default defconfig file, we need to reconfigure.

For the convenience of everyone's configuration, please refer to the following table:

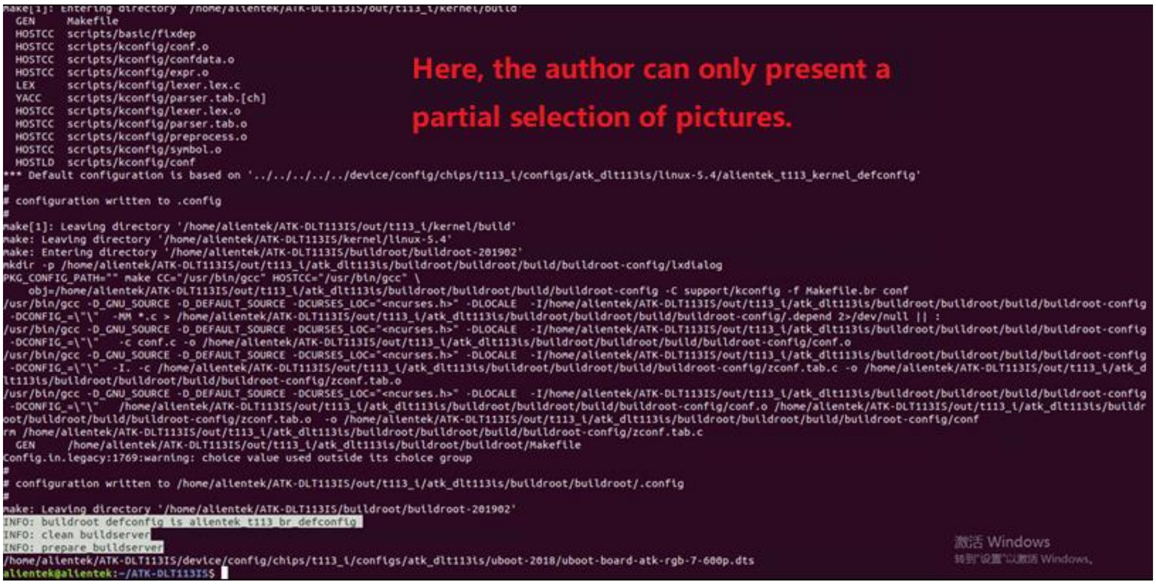
Storage mode	System	Screen	Select a number
	Buildroot	RGB 7 inch 1024x600	111000
		LVDS 10.1 inch 1280x800	111010
		LVDS 8-lane display (tested and developed)	111020
		Cvbs interface - Purchase a conversion interface yourself	111030
		MIPI 5.5 inch 720x1280	111040
		MIPI 5.5 inch 1080x1920	111050
		MIPI 10.1 inch 800x1280	111060
		RGB 4.3 inch 800x480	111070
		RGB 7 inch 800x480	111080
		RGB 10.1 inch 1280x800	111090
		RGB 7 inch 1024x600	121000
		LVDS 10.1 inch 1280x800	121010
		LVDS 8-lane display (tested and developed)	121020
		Cvbs interface - Purchase a conversion interface yourself	121030

eMMC	openwrt	MIPI 5.5 inch 720x1280	121040
		MIPI 5.5 inch 1080x1920	121050
		MIPI 10.1 inch 800x1280	121060
		RGB 4.3 inch 800x480	121070
		RGB 7 inch 800x480	121080
		RGB 10.1 inch 1280x800	121090
NAND	Buildroot	RGB 7 inch 1024x600	111100
		LVDS 10.1 inch 1280x800	111110
		LVDS 8-lane display (tested and developed)	111120
		Cvbs interface - Purchase a conversion interface yourself	111130
		MIPI 5.5 inch 720x1280	111140
		MIPI 5.5 inch 1080x1920	111150
		MIPI 10.1 inch 800x1280	111160
		RGB 4.3 inch 800x480	111170
		RGB 7 inch 800x480	111180
		RGB 10.1 inch 1280x800	111190
	openwrt	RGB 7 inch 1024x600	121100
		LVDS 10.1 inch 1280x800	121110
		LVDS 8-lane display (tested and developed)	121120
		Cvbs interface - Purchase a conversion interface yourself	121130
		MIPI 5.5 inch 720x1280	121140
		MIPI 5.5 inch 1080x1920	121150
		MIPI 10.1 inch 800x1280	121160
		RGB 4.3 inch 800x480	121170

		RGB 7 inch 800x480	121180
		RGB 10.1 inch 1280x800	121190

Note: The nand version of openwrt here cannot be compiled successfully. Don't bother asking us. The SDK provided by Allwinner also cannot be compiled.

After the configuration is completed, the following output information will be displayed as shown in the figure below:



```
make[1]: Entering directory '/home/alientek/ATK-DLT113IS/out/t113_i/kernel/build'
  GEN     Makefile
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/conf.o
  HOSTCC  scripts/kconfig/confdata.o
  HOSTCC  scripts/kconfig/expr.o
  LEX     scripts/kconfig/lexer.lex.c
  YACC    scripts/kconfig/parser.tab.[ch]
  HOSTCC  scripts/kconfig/lexer.lex.o
  HOSTCC  scripts/kconfig/parser.tab.o
  HOSTCC  scripts/kconfig/preprocess.o
  HOSTCC  scripts/kconfig/symbol.o
  HOSTLD  scripts/kconfig/conf
*** Default configuration is based on '../../../../../device/config/chips/t113_i/configs/atk_dlt113is/linux-5.4/alientek_t113_kernel_defconfig'
#
# configuration written to .config
#
make[1]: Leaving directory '/home/alientek/ATK-DLT113IS/out/t113_i/kernel/build'
make: Leaving directory '/home/alientek/ATK-DLT113IS/kernel/linux-5.4'
make: Entering directory '/home/alientek/ATK-DLT113IS/buildroot/buildroot-201902'
mkdir -p /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/buildroot-config/lxdialog
PKG_CONFIG_PATH="" make CC="/usr/bin/gcc" HOSTCC="/usr/bin/gcc" \
    obj=/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/buildroot-config -C support/kconfig -f Makefile.br conf
/usr/bin/gcc -D_GNU_SOURCE -D_DEFAULT_SOURCE -DCURSES_LOC="<ncurses.h>" -DLOCALE  -I/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/buildroot-config
 -DCONFIG_=\"\"  -MM *.c > /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/buildroot-config/.depend 2>/dev/null || :
/usr/bin/gcc -D_GNU_SOURCE -D_DEFAULT_SOURCE -DCURSES_LOC="<ncurses.h>" -DLOCALE  -I/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/buildroot-config
 -DCONFIG_=\"\"   -c conf.c -o /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/buildroot-config/conf.o
/usr/bin/gcc -D_GNU_SOURCE -D_DEFAULT_SOURCE -DCURSES_LOC="<ncurses.h>" -DLOCALE  -I/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/buildroot-config
 -DCONFIG_=\"\"  -I. -c /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/buildroot-config/zconf.tab.c -o /home/alientek/ATK-DLT113IS/out/t113_i/atk_d
lt113is/buildroot/buildroot/build/buildroot-config/zconf.tab.o
/usr/bin/gcc -D_GNU_SOURCE -D_DEFAULT_SOURCE -DCURSES_LOC="<ncurses.h>" -DLOCALE  -I/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/buildroot-config
 -DCONFIG_=\"\"   /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/buildroot-config/conf.o /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildr
oot/buildroot/build/buildroot-config/zconf.tab.o  -o /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/buildroot-config/conf
rm /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/buildroot-config/zconf.tab.c
  GEN     /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/Makefile
Config.in.legacy:1769:warning: choice value used outside its choice group
#
# configuration written to /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/.config
#
make: Leaving directory '/home/alientek/ATK-DLT113IS/buildroot/buildroot-201902'
INFO: buildroot defconfig is alientek_t113_br_defconfig
INFO: clean buildserver
INFO: prepare buildserver
/home/alientek/ATK-DLT113IS/device/config/chips/t113_i/configs/atk_dlt113is/uboot-2018/uboot-board-atk-rgb-7-600p.dts
alientek@alientek:~/ATK-DLT113IS$
```

Here, the author can only present a partial selection of pictures.

Figure 1.3-8 Post-configuration printout

After configuration is completed, a hidden file named ".buildconfig" will be generated in the source code directory. The content of the file is as follows:

Part of the content of the .buildconfig file in the sample code 1.3.1

```
export LICHEE_PLATFORM=linux
export LICHEE_LINUX_DEV=buildroot
export LICHEE_IC=t113_i
export LICHEE_BOARD=atk_dlt113is
export LICHEE_BRANDY_DTS=atk-rgb-7-600p
export LICHEE_FLASH=default
export LICHEE_KERNEL_ARCH=arm
export LICHEE_ARCH=arm
export LICHEE_KERN_VER=linux-5.4
export LICHEE_KERNEL_VERSION=5.4.61
export LICHEE_KERN_DEFCONF=alientek_t113_kernel_defconfig
export LICHEE_KERN_DEFCONF_RT=
export LICHEE_ATK_KERN_COMMON=board-atk-common.dtsi
export LICHEE_BUILDING_SYSTEM=buildroot
export LICHEE_BR_VER=201902
export LICHEE_BR_DEFCONF=alientek_t113_br_defconfig
export LICHEE_DEFCONFIG_FRAGMENT=
export LICHEE_PRODUCT=t113_evb1_auto
export LICHEE_BRANDY_VER=2.0
```

```
20 export LICHEE_BRANDY_DEFCONF=alientek_t113_uboot_defconfig
21 export LICHEE_BRANDY_UBOOT_VER=2018
22 export LICHEE_COMPILER_TAR=arm/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi.tar.xz
23 export LICHEE_ROOTFS=target-arm-linaro-5.3.tar.bz2
24 export LICHEE_BUSSINESS=
25 export LICHEE_BR_RAMFS_CONF=
26 export LICHEE_CHIP=sun8iw20p1
27 export LICHEE_RTOS_PROJECT_NAME=t113_i_c906_evb1_auto
28 export LICHEE_DSP_PROJECT_NAME=
29 export LICHEE_PACK_HOOK=
30 export LICHEE_REDUNDANT_ENV_SIZE=0x20000
31 export LICHEE_BRANDY_SPL=
32 export LICHEE_COMPRESS=gzip
33 export LICHEE_NO_RAMDISK_NEEDED=y
34 export LICHEE_RAMDISK_PATH=
35 export LICHEE_KERN_DEFCONF_RECOVERY=config-5.4-recovery
36 export LICHEE_USE_INDEPENDENT_BSP=
37 export LICHEE_INDEPENDENT_PACK=
38 export LICHEE_BOOT0_BIN_NAME=
39 export LICHEE_EFEX_BIN_NAME=
40 export ANDROID_CLANG_PATH=
41 export ANDROID_TOOLCHAIN_PATH=
42 export ANDROID_CLANG_ARGS=
43 export LICHEE_BSP_STAGING=
44 export LICHEE_KERN_SYSTEM=
45 export LICHEE_KERN_DEFCONF_RELATIVE=../../../../../device/config/chips/t113_i/configs/atk_dlt113is/linux-5.4/alientek_t113_kernel_defconfig
46 export LICHEE_KERN_DEFCONF_ABSOLUTE=/home/alientek/ATK-DLT113IS/device/config/chips/t113_i/configs/atk_dlt113is/linux-5.4/alientek_t113_kernel_defconfig
```

From the example code 1.3.1, it can be seen that the important content is based on the atk_dlt113is folder. Next, we will briefly explain the content of this file:

Line 1: Set the platform as Linux

Line 2: File system as buildroot

Line 3: Chip model as t113_i

Line 4: Board-level configuration as atk_dlt113is

Line 5: Select different device trees based on the screen. For example, if the screen is: rgb 7-inch 1024x600, then select atk-rgb-7-600p

Line 6: flash selection as default

Line 7: Kernel architecture as arm

Line 10: Kernel version as 5.4.61

Line 11: Kernel defconfig file as alientek_t113_kernel_defconfig

Line 13: Pynq-Arduino custom variable for the general device tree as board-atk-common.dtsi

Line 15: Buildroot version as 201902

Line 16: Buildroot defconfig file as alientek_t113_br_defconfig

Line 20: Uboot defconfig file as alientek_t113_uboot_defconfig

Line 21: Uboot version as 2018

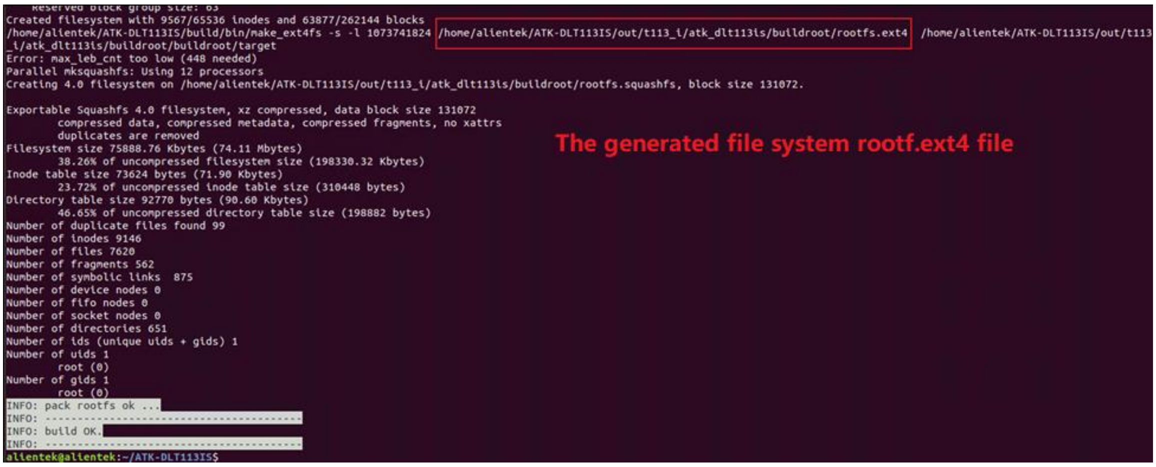
Other information can be viewed by yourself.

1.3.2 Full Compilation

Full compilation is quite simple. The prerequisite is that the file ".buildconfig" must exist in the SDK source code directory. The command is as follows:

```
./build.sh config // Configured, no need to configure again
./build.sh            // Direct compilation
```

If no configuration is set, direct compilation will result in a compilation error (when performing full compilation and there is no .buildconfig file, the config configuration will be automatically run, which will cause a compilation error).



```
    Reserved block group size: 63
Created filesystem with 9567/65536 inodes and 63877/262144 blocks
/home/alientek/ATK-DLT113IS/build/bin/make_ext4fs -s -l 1073741824 /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/rootfs.ext4  /home/alientek/ATK-DLT113IS/out/t113
_i/atk_dlt113is/buildroot/buildroot/target
Error: max_leb_cnt too low (448 needed)
Parallel mksquashfs: Using 12 processors
Creating 4.0 filesystem on /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/rootfs.squashfs, block size 131072.

Exportable Squashfs 4.0 filesystem, xz compressed, data block size 131072
        compressed data, compressed metadata, compressed fragments, no xattrs
        duplicates are removed
Filesystem size 75888.76 Kbytes (74.11 Mbytes)
        38.26% of uncompressed filesystem size (198330.32 Kbytes)
Inode table size 73624 bytes (71.90 Kbytes)
        23.72% of uncompressed inode table size (310448 bytes)
Directory table size 92770 bytes (90.60 Kbytes)
        46.65% of uncompressed directory table size (198882 bytes)
Number of duplicate files found 99
Number of inodes 9146
Number of files 7620
Number of fragments 562
Number of symbolic links  875
Number of device nodes 0
Number of fifo nodes 0
Number of socket nodes 0
Number of directories 651
Number of ids (unique uids + gids) 1
Number of uids 1
        root (0)
Number of gids 1
        root (0)
INFO: pack rootfs ok ...
INFO: ----------------------------------------
INFO: build OK.
INFO: ----------------------------------------
alientek@alientek:~/ATK-DLT113IS$
```

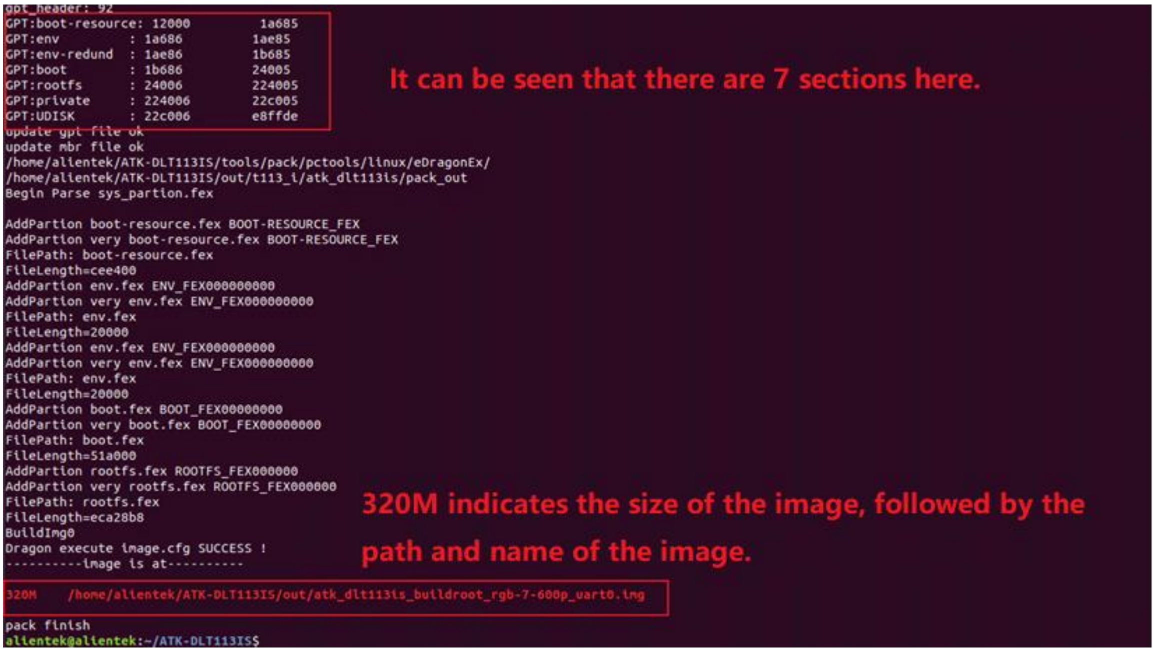
The generated file system rootf.ext4 file

Figure 1.3-9 Compilation completed print information

1.3.3 Image packaging

In 1.3.2, the compilation was completed, but an img image needs to be generated. The official burn-in image of Allwinner requires the img format, so the following code is run to package the image:

```
./build.sh pack
```



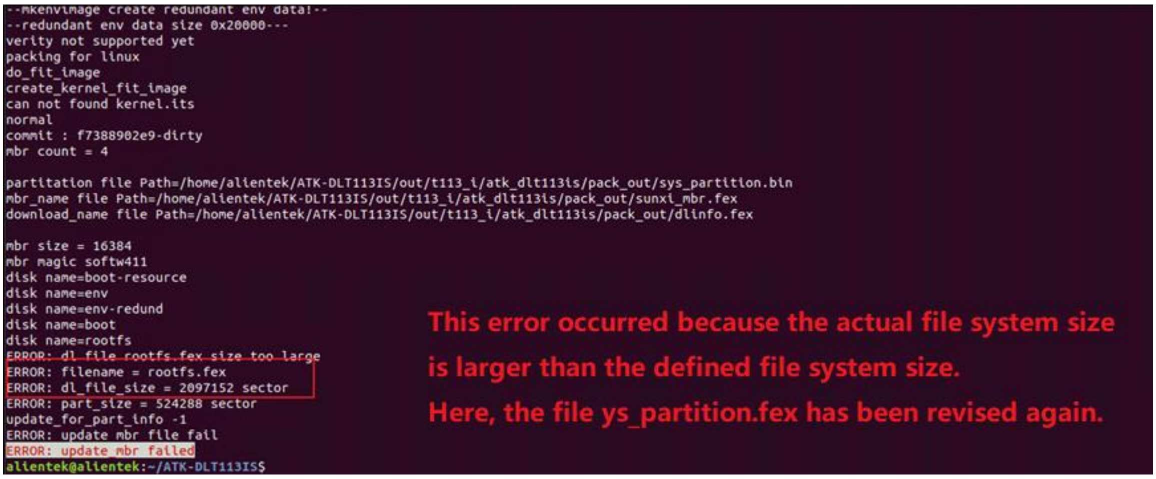
```
gpt_header: 92
GPT:boot-resource: 12000         1a685
GPT:env          : 1a686         1ae85
GPT:env-redund   : 1ae86         1b685
GPT:boot         : 1b686         24005
GPT:rootfs       : 24006         224005
GPT:private      : 224006        22c005
GPT:UDISK        : 22c006        e8ffde
update gpt file ok
update mbr file ok
/home/alientek/ATK-DLT113IS/tools/pack/pctools/linux/eDragonEx/
/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/pack_out
Begin Parse sys_partion.fex

AddPartion boot-resource.fex BOOT-RESOURCE_FEX
AddPartion very boot-resource.fex BOOT-RESOURCE_FEX
FilePath: boot-resource.fex
FileLength=cee400
AddPartion env.fex ENV_FEX000000000
AddPartion very env.fex ENV_FEX000000000
FilePath: env.fex
FileLength=20000
AddPartion env.fex ENV_FEX000000000
AddPartion very env.fex ENV_FEX000000000
FilePath: env.fex
FileLength=20000
AddPartion boot.fex BOOT_FEX00000000
AddPartion very boot.fex BOOT_FEX00000000
FilePath: boot.fex
FileLength=51a000
AddPartion rootfs.fex ROOTFS_FEX000000
AddPartion very rootfs.fex ROOTFS_FEX000000
FilePath: rootfs.fex
FileLength=eca28b8
BuildImg0
Dragon execute image.cfg SUCCESS !
----------image is at----------

320M    /home/alientek/ATK-DLT113IS/out/atk_dlt113is_buildroot_rgb-7-600p_uart0.img

pack finish
alientek@alientek:~/ATK-DLT113IS$
```

It can be seen that there are 7 sections here.

320M indicates the size of the image, followed by the path and name of the image.

Figure 1.3-10 The "pack" command is used for packaging.

After the packaging is completed, the information shown in the above figure will be output. Here, a 320M image is generated (the size is determined according to the configuration in the buildroot and depends on the released buildroot). The "atk_dlt113is_buildroot_rgb-7-600p_uart0.img" image is an emmc version of the buildroot image and only supports the screen of RGB-7-inch-1024x600.



```
--mkenvimage create redundant env data!--
--redundant env data size 0x20000---
verity not supported yet
packing for linux
do_fit_image
create_kernel_fit_image
can not found kernel.its
normal
commit : f7388902e9-dirty
mbr count = 4

partitation file Path=/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/pack_out/sys_partition.bin
mbr_name file Path=/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/pack_out/sunxi_mbr.fex
download_name file Path=/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/pack_out/dlinfo.fex

mbr size = 16384
mbr magic softw411
disk name=boot-resource
disk name=env
disk name=env-redund
disk name=boot
disk name=rootfs
ERROR: dl_file rootfs.fex size too large
ERROR: filename = rootfs.fex
ERROR: dl_file_size = 2097152 sector
ERROR: part_size = 524288 sector
update_for_part_info -1
ERROR: update mbr file fail
ERROR: update mbr failed
alientek@alientek:~/ATK-DLT113IS$
```

This error occurred because the actual file system size is larger than the defined file system size.
Here, the file ys_partition.fex has been revised again.

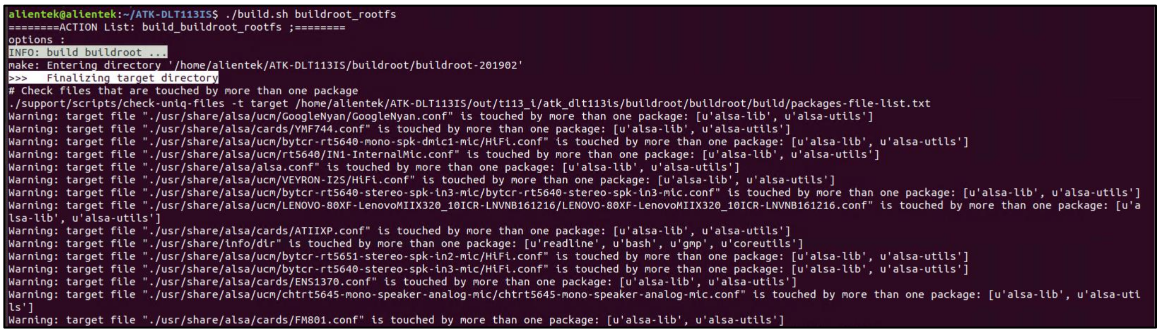
Figure 1.3-11 There is a packaging error.

Here are just the errors that I encountered. You should modify the sys_partition.fex file by yourselves. If there are any errors, please handle them by yourself.

1.3.4 File System Compilation

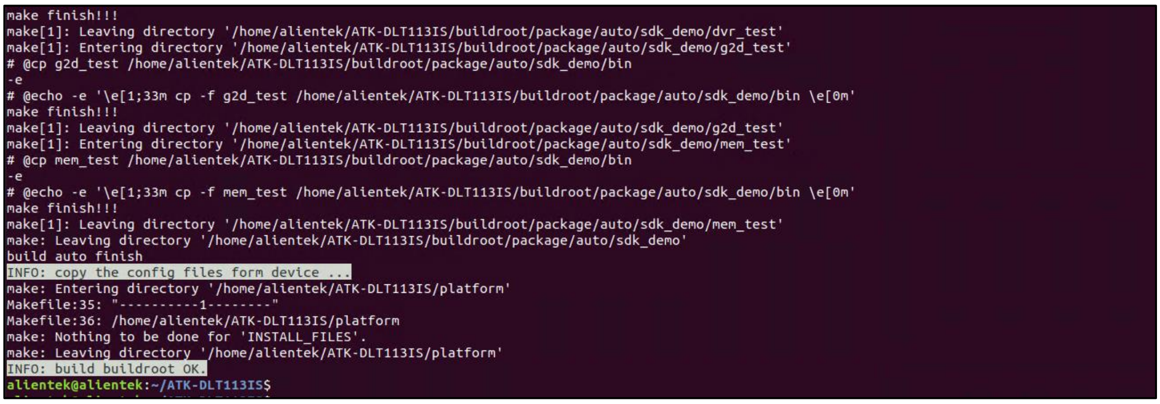
Compile the buildroot file system separately. Run the following command:

```
./build.sh buildroot_rootfs
```



```
alientek@alientek:~/ATK-DLT113IS$ ./build.sh buildroot_rootfs
========ACTION List: build_buildroot_rootfs ;========
options :
INFO: build buildroot ...
make: Entering directory '/home/alientek/ATK-DLT113IS/buildroot/buildroot-201902'
>>>   Finalizing target directory
# Check files that are touched by more than one package
./support/scripts/check-uniq-files -t target /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/packages-file-list.txt
Warning: target file "./usr/share/alsa/ucm/GoogleNyan/GoogleNyan.conf" is touched by more than one package: [u'alsa-lib', u'alsa-utils']
Warning: target file "./usr/share/alsa/cards/YMF744.conf" is touched by more than one package: [u'alsa-lib', u'alsa-utils']
Warning: target file "./usr/share/alsa/ucm/bytcr-rt5640-mono-spk-dmic1-mic/HiFi.conf" is touched by more than one package: [u'alsa-lib', u'alsa-utils']
Warning: target file "./usr/share/alsa/ucm/rt5640/IN1-InternalMic.conf" is touched by more than one package: [u'alsa-lib', u'alsa-utils']
Warning: target file "./usr/share/alsa/alsa.conf" is touched by more than one package: [u'alsa-lib', u'alsa-utils']
Warning: target file "./usr/share/alsa/ucm/VEYRON-I2S/HiFi.conf" is touched by more than one package: [u'alsa-lib', u'alsa-utils']
Warning: target file "./usr/share/alsa/ucm/bytcr-rt5640-stereo-spk-in3-mic/bytcr-rt5640-stereo-spk-in3-mic.conf" is touched by more than one package: [u'alsa-lib', u'alsa-utils']
Warning: target file "./usr/share/alsa/ucm/LENOVO-80XF-LenovoMIIX320_10ICR-LNVNB161216/LENOVO-80XF-LenovoMIIX320_10ICR-LNVNB161216.conf" is touched by more than one package: [u'a
lsa-lib', u'alsa-utils']
Warning: target file "./usr/share/alsa/cards/ATIIXP.conf" is touched by more than one package: [u'alsa-lib', u'alsa-utils']
Warning: target file "./usr/share/info/dir" is touched by more than one package: [u'readline', u'bash', u'gmp', u'coreutils']
Warning: target file "./usr/share/alsa/ucm/bytcr-rt5651-stereo-spk-in2-mic/HiFi.conf" is touched by more than one package: [u'alsa-lib', u'alsa-utils']
Warning: target file "./usr/share/alsa/ucm/bytcr-rt5640-stereo-spk-in3-mic/HiFi.conf" is touched by more than one package: [u'alsa-lib', u'alsa-utils']
Warning: target file "./usr/share/alsa/cards/ENS1370.conf" is touched by more than one package: [u'alsa-lib', u'alsa-utils']
Warning: target file "./usr/share/alsa/ucm/chtrt5645-mono-speaker-analog-mic/chtrt5645-mono-speaker-analog-mic.conf" is touched by more than one package: [u'alsa-lib', u'alsa-uti
ls']
Warning: target file "./usr/share/alsa/cards/FM801.conf" is touched by more than one package: [u'alsa-lib', u'alsa-utils']
```

Figure 1.3-12 Compile separately buildroot

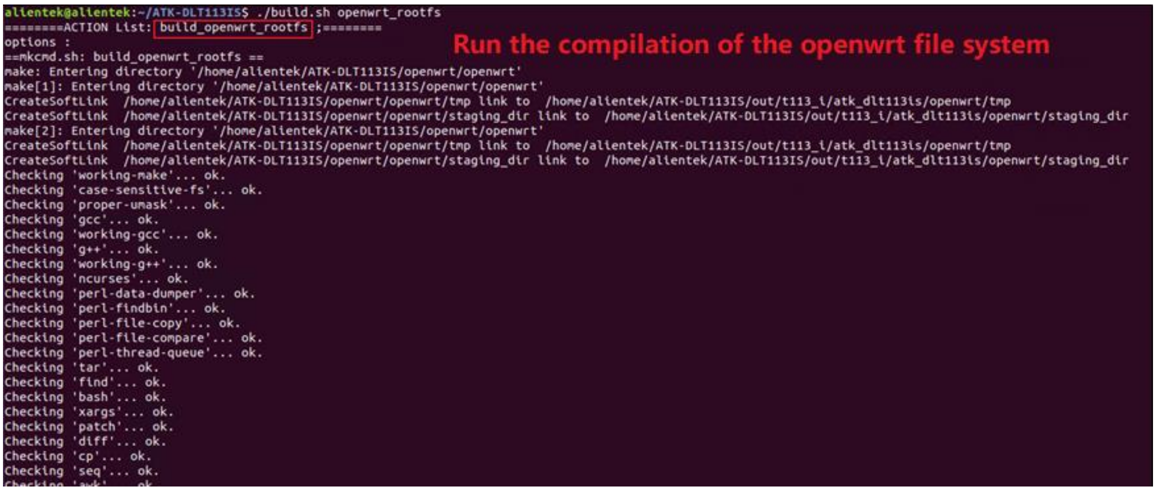


```
make finish!!!
make[1]: Leaving directory '/home/alientek/ATK-DLT113IS/buildroot/package/auto/sdk_demo/dvr_test'
make[1]: Entering directory '/home/alientek/ATK-DLT113IS/buildroot/package/auto/sdk_demo/g2d_test'
# @cp g2d_test /home/alientek/ATK-DLT113IS/buildroot/package/auto/sdk_demo/bin
-e
# @echo -e '\e[1;33m cp -f g2d_test /home/alientek/ATK-DLT113IS/buildroot/package/auto/sdk_demo/bin \e[0m'
make finish!!!
make[1]: Leaving directory '/home/alientek/ATK-DLT113IS/buildroot/package/auto/sdk_demo/g2d_test'
make[1]: Entering directory '/home/alientek/ATK-DLT113IS/buildroot/package/auto/sdk_demo/mem_test'
# @cp mem_test /home/alientek/ATK-DLT113IS/buildroot/package/auto/sdk_demo/bin
-e
# @echo -e '\e[1;33m cp -f mem_test /home/alientek/ATK-DLT113IS/buildroot/package/auto/sdk_demo/bin \e[0m'
make finish!!!
make[1]: Leaving directory '/home/alientek/ATK-DLT113IS/buildroot/package/auto/sdk_demo/mem_test'
make: Leaving directory '/home/alientek/ATK-DLT113IS/buildroot/package/auto/sdk_demo'
build auto finish
INFO: copy the config files form device ...
make: Entering directory '/home/alientek/ATK-DLT113IS/platform'
Makefile:35: "----------1--------"
Makefile:36: /home/alientek/ATK-DLT113IS/platform
make: Nothing to be done for 'INSTALL_FILES'.
make: Leaving directory '/home/alientek/ATK-DLT113IS/platform'
INFO: build buildroot OK.
alientek@alientek:~/ATK-DLT113IS$
```

Figure 1.3-13 Buildroot compilation is successful

If you choose openwrt for configuration, the compilation command is as follows (you must compile everything at once. When packaging openwrt, some drivers of the kernel are required. If there is an error message "ERROR: target/linux failed to build", you can only delete the entire out/* directory and recompile everything from scratch):

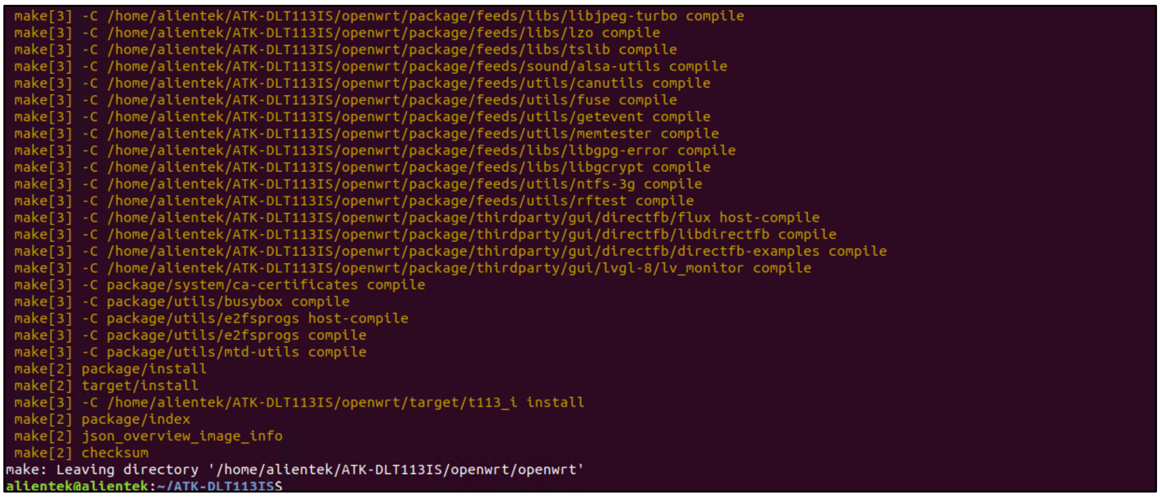
```
./build.sh openwrt_rootfs
```



Run the compilation of the openwrt file system

```
alientek@alientek:~/ATK-DLT113IS$ ./build.sh openwrt_rootfs
========ACTION List: build_openwrt_rootfs ;========
options :
==mkcmd.sh: build_openwrt_rootfs ==
make: Entering directory '/home/alientek/ATK-DLT113IS/openwrt/openwrt'
make[1]: Entering directory '/home/alientek/ATK-DLT113IS/openwrt/openwrt'
CreateSoftLink  /home/alientek/ATK-DLT113IS/openwrt/openwrt/tmp link to  /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/tmp
CreateSoftLink  /home/alientek/ATK-DLT113IS/openwrt/openwrt/staging_dir link to  /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/staging_dir
make[2]: Entering directory '/home/alientek/ATK-DLT113IS/openwrt/openwrt'
CreateSoftLink  /home/alientek/ATK-DLT113IS/openwrt/openwrt/tmp link to  /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/tmp
CreateSoftLink  /home/alientek/ATK-DLT113IS/openwrt/openwrt/staging_dir link to  /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/staging_dir
Checking 'working-make'... ok.
Checking 'case-sensitive-fs'... ok.
Checking 'proper-umask'... ok.
Checking 'gcc'... ok.
Checking 'working-gcc'... ok.
Checking 'g++'... ok.
Checking 'working-g++'... ok.
Checking 'ncurses'... ok.
Checking 'perl-data-dumper'... ok.
Checking 'perl-findbin'... ok.
Checking 'perl-file-copy'... ok.
Checking 'perl-file-compare'... ok.
Checking 'perl-thread-queue'... ok.
Checking 'tar'... ok.
Checking 'find'... ok.
Checking 'bash'... ok.
Checking 'xargs'... ok.
Checking 'patch'... ok.
Checking 'diff'... ok.
Checking 'cp'... ok.
Checking 'seq'... ok.
Checking 'awk'... ok.
```

Figure 1.3-14 Compile separately openwrt system



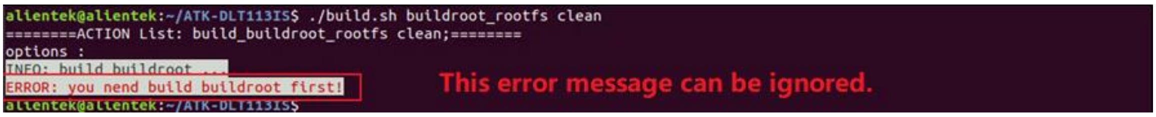
```
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/feeds/libs/libjpeg-turbo compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/feeds/libs/lzo compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/feeds/libs/tslib compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/feeds/sound/alsa-utils compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/feeds/utils/canutils compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/feeds/utils/fuse compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/feeds/utils/getevent compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/feeds/utils/memtester compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/feeds/libs/libgpg-error compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/feeds/libs/libgcrypt compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/feeds/utils/ntfs-3g compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/feeds/utils/rftest compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/thirdparty/gui/directfb/flux host-compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/thirdparty/gui/directfb/libdirectfb compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/thirdparty/gui/directfb/directfb-examples compile
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/package/thirdparty/gui/lvgl-8/lv_monitor compile
 make[3] -C package/system/ca-certificates compile
 make[3] -C package/utils/busybox compile
 make[3] -C package/utils/e2fsprogs host-compile
 make[3] -C package/utils/e2fsprogs compile
 make[3] -C package/utils/mtd-utils compile
 make[2] package/install
 make[2] target/install
 make[3] -C /home/alientek/ATK-DLT113IS/openwrt/target/t113_i install
 make[2] package/index
 make[2] json_overview_image_info
 make[2] checksum
make: Leaving directory '/home/alientek/ATK-DLT113IS/openwrt/openwrt'
alientek@alientek:~/ATK-DLT113IS$
```

Figure 1.3-15 Compilation completed

1.3.5 File system cleanup

To separately clear the compilation of buildroot, run the following command to perform the file system cleanup:

```
./build.sh buildroot_rootfs clean
```



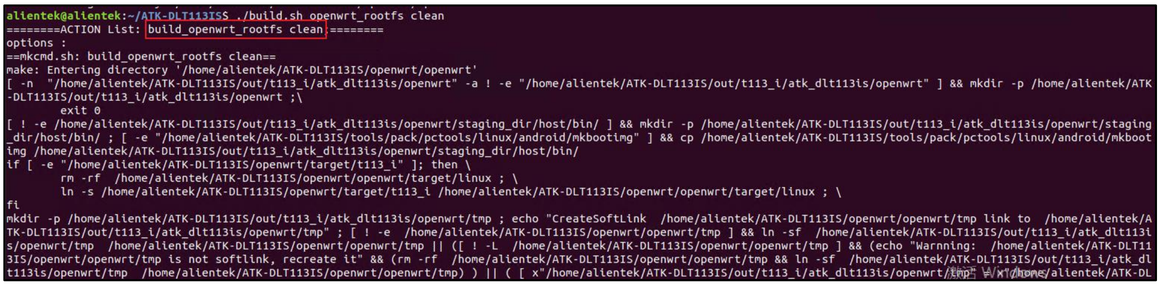
```
alientek@alientek:~/ATK-DLT113IS$ ./build.sh buildroot_rootfs clean
========ACTION List: build_buildroot_rootfs clean;========
options :
INFO: build buildroot ...
ERROR: you nend build buildroot first!
alientek@alientek:~/ATK-DLT113IS$
```

This error message can be ignored.

Figure 1.3-16 Clearing Buildroot

To completely remove the compilation of openwrt, run the following command to clear the file system:

```
./build.sh openwrt_rootfs clean
```



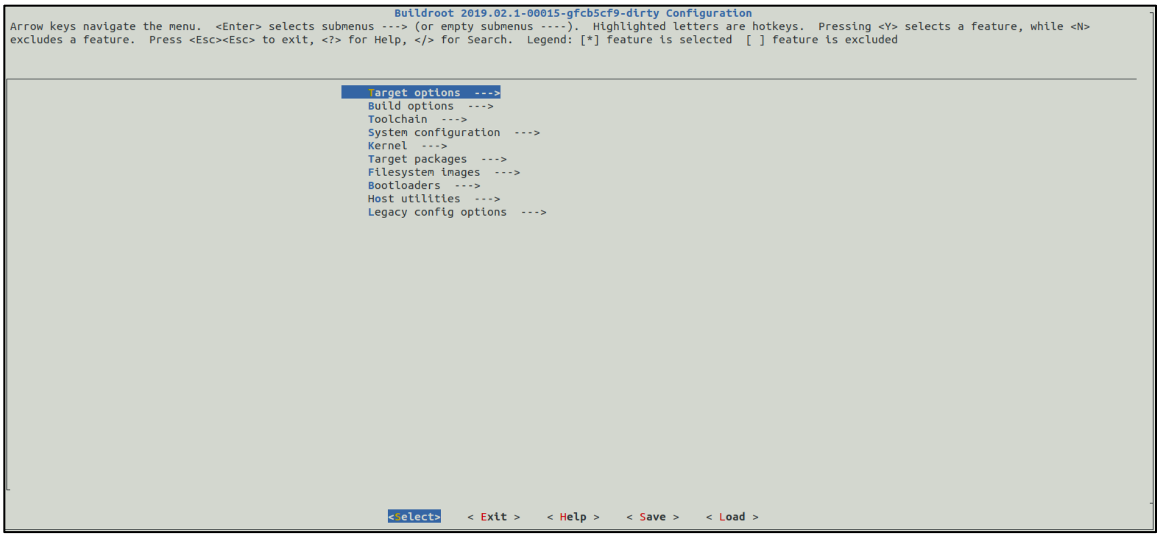
```
alientek@alientek:~/ATK-DLT113IS$ ./build.sh openwrt_rootfs clean
========ACTION List: build_openwrt_rootfs clean;========
options :
==mkcmd.sh: build_openwrt_rootfs clean==
make: Entering directory '/home/alientek/ATK-DLT113IS/openwrt/openwrt'
[ -n  "/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt" -a ! -e "/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt" ] && mkdir -p /home/alientek/ATK
-DLT113IS/out/t113_i/atk_dlt113is/openwrt ;\
        exit 0
[ ! -e /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/staging_dir/host/bin/ ] && mkdir -p /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/staging
_dir/host/bin/ ; [ -e "/home/alientek/ATK-DLT113IS/tools/pack/pctools/linux/android/mkbootimg" ] && cp /home/alientek/ATK-DLT113IS/tools/pack/pctools/linux/android/mkboot
img /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/staging_dir/host/bin/
if [ -e "/home/alientek/ATK-DLT113IS/openwrt/target/t113_i" ]; then \
        rm -rf  /home/alientek/ATK-DLT113IS/openwrt/openwrt/target/linux ; \
        ln -s /home/alientek/ATK-DLT113IS/openwrt/target/t113_i /home/alientek/ATK-DLT113IS/openwrt/openwrt/target/linux ; \
fi
mkdir -p /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/tmp ; echo "CreateSoftLink  /home/alientek/ATK-DLT113IS/openwrt/openwrt/tmp link to  /home/alientek/A
TK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/tmp" ; [ ! -e  /home/alientek/ATK-DLT113IS/openwrt/openwrt/tmp ] && ln -sf  /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113i
s/openwrt/tmp  /home/alientek/ATK-DLT113IS/openwrt/openwrt/tmp || ([ ! -L  /home/alientek/ATK-DLT113IS/openwrt/openwrt/tmp ] && (echo "Warnning:  /home/alientek/ATK-DLT11
3IS/openwrt/openwrt/tmp is not softlink, recreate it" && (rm -rf  /home/alientek/ATK-DLT113IS/openwrt/openwrt/tmp && ln -sf  /home/alientek/ATK-DLT113IS/out/t113_i/atk_dl
t113is/openwrt/tmp  /home/alientek/ATK-DLT113IS/openwrt/openwrt/tmp) ) || ( [ x"/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/tmp" = x"/home/alientek/ATK-DL
```

Figure 1.3-17 Clearing the logs of openwrt

1.3.6 Configuration of Buildroot

First, configure it as a buildroot system. Then, run the following command to configure buildroot:

```
./build.sh buildroot_menuconfig
```



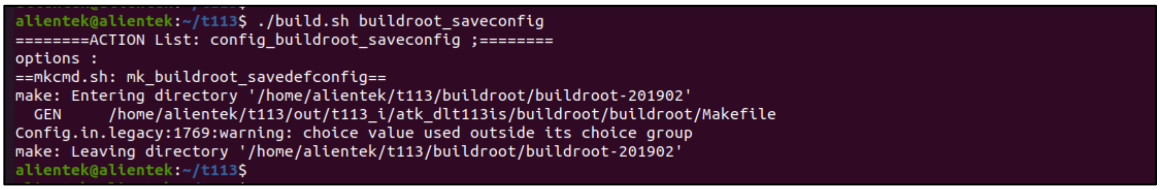
```
                 Buildroot 2019.02.1-00015-gfcb5cf9-dirty Configuration
Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> selects a feature, while <N>
excludes a feature.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] feature is selected  [ ] feature is excluded

                    Target options  --->
                    Build options  --->
                    Toolchain  --->
                    System configuration  --->
                    Kernel  --->
                    Target packages  --->
                    Filesystem images  --->
                    Bootloaders  --->
                    Host utilities  --->
                    Legacy config options  --->

              <Select>    < Exit >    < Help >    < Save >    < Load >
```

Figure 1.3-18 Enter the menuconfig interface

Note: In this buildroot configuration interface, the default configuration will be called from the "out" directory. The modifications are only to the output files and cannot directly modify the alientek_t113_br_defconfig.

After the configuration is completed, you can run the following command to save it.

```
./build.sh buildroot_saveconfig
```



```
alientek@alientek:~/t113$ ./build.sh buildroot_saveconfig
========ACTION List: config_buildroot_saveconfig ;========
options :
==mkcmd.sh: mk_buildroot_savedefconfig==
make: Entering directory '/home/alientek/t113/buildroot/buildroot-201902'
  GEN     /home/alientek/t113/out/t113_i/atk_dlt113is/buildroot/buildroot/Makefile
Config.in.legacy:1769:warning: choice value used outside its choice group
make: Leaving directory '/home/alientek/t113/buildroot/buildroot-201902'
alientek@alientek:~/t113$
```

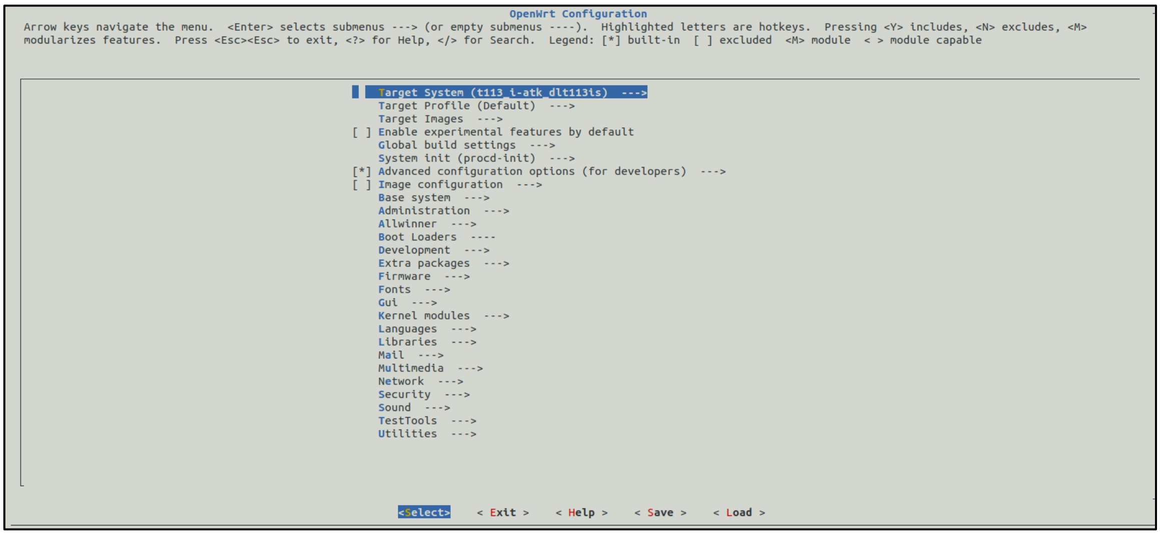
Figure 1.3-19 Save the configuration of buildroot

By default, the ".config" file in the "out" directory will be saved as "buildroot/buildroot-201902/configs/alientek_t113_br_defconfig".

1.3.7 Configuration of openwrt

Openwrt can only be configured but not saved. Therefore, we can only directly modify the configuration file. First, open the default configuration and run the following command:

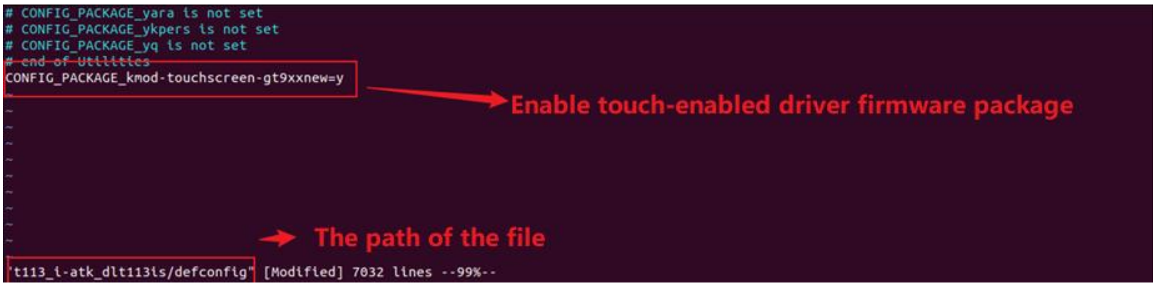
```
./build.sh openwrt_menuconfig
```



```
                                   OpenWrt Configuration
Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M>
modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                          Target System (t113_i-atk_dlt113is)  --->
                          Target Profile (Default)  --->
                          Target Images  --->
                      [ ] Enable experimental features by default
                          Global build settings  --->
                          System init (procd-init)  --->
                      [*] Advanced configuration options (for developers)  --->
                      [ ] Image configuration  --->
                          Base system  --->
                          Administration  --->
                          Allwinner  --->
                          Boot Loaders  ----
                          Development  --->
                          Extra packages  --->
                          Firmware  --->
                          Fonts  --->
                          Gui  --->
                          Kernel modules  --->
                          Languages  --->
                          Libraries  --->
                          Mail  --->
                          Multimedia  --->
                          Network  --->
                          Security  --->
                          Sound  --->
                          TestTools  --->
                          Utilities  --->

                    <Select>    < Exit >    < Help >    < Save >    < Load >
```

Figure 1.3-20 openwrt Configuration

It is impossible to save the configuration file to the defconfig file using commands (it can only be saved to the compiled output file. Deleting the out folder will result in the loss of all your configurations). To save the configuration, simply modify the defconfig file directly. Locate the configuration item you want to enable, and press the "h" key to view the corresponding macro definition. For example: To enable the "CONFIG_PACKAGE_kmod-touchscreen-gt9xxnew" touch firmware package, remember this macro. Directly modify the "openwrt/target/t113_i/t113_i-atk_dlt113is/defconfig" file in the SDK's source code directory. Write this macro into the defconfig file. For the NAND version, the defconfig file is "openwrt/target/t113_i/t113_i-atk_dlt113is_nand/defconfig".



```
# CONFIG_PACKAGE_yara is not set
# CONFIG_PACKAGE_ykpers is not set
# CONFIG_PACKAGE_yq is not set
# end of Utilities
CONFIG_PACKAGE_kmod-touchscreen-gt9xxnew=y
~
~
~
~
~
~
~
~
~
~
"t113_i-atk_dlt113is/defconfig" [Modified] 7032 lines --99%--
```

Enable touch-enabled driver firmware package

The path of the file

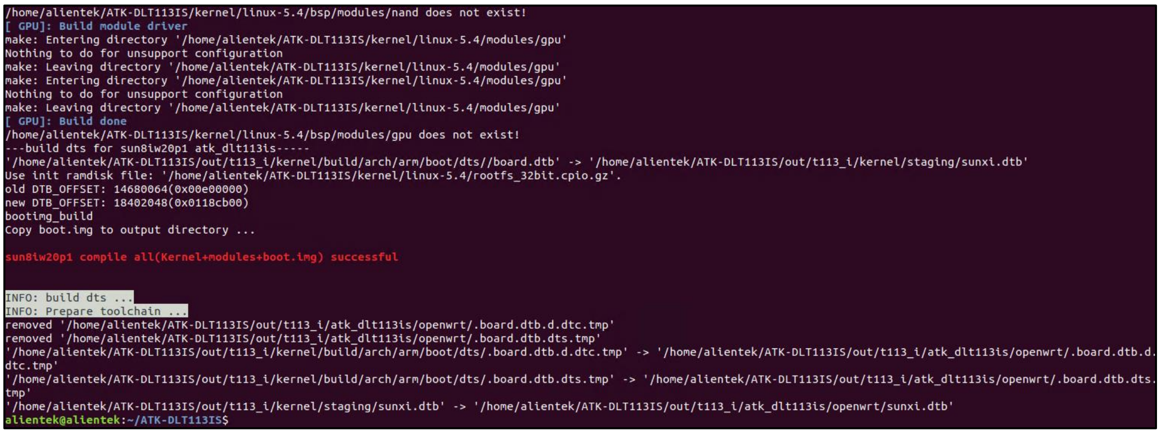
Figure 1.3-21 Save the openwrt configuration

After the saving is completed, by running "./build.sh config" again, you will be able to see that this touch firmware has been enabled.

1.3.8 Kernel Compilation

Run the following command to perform the compilation:

```
./build.sh kernel
```



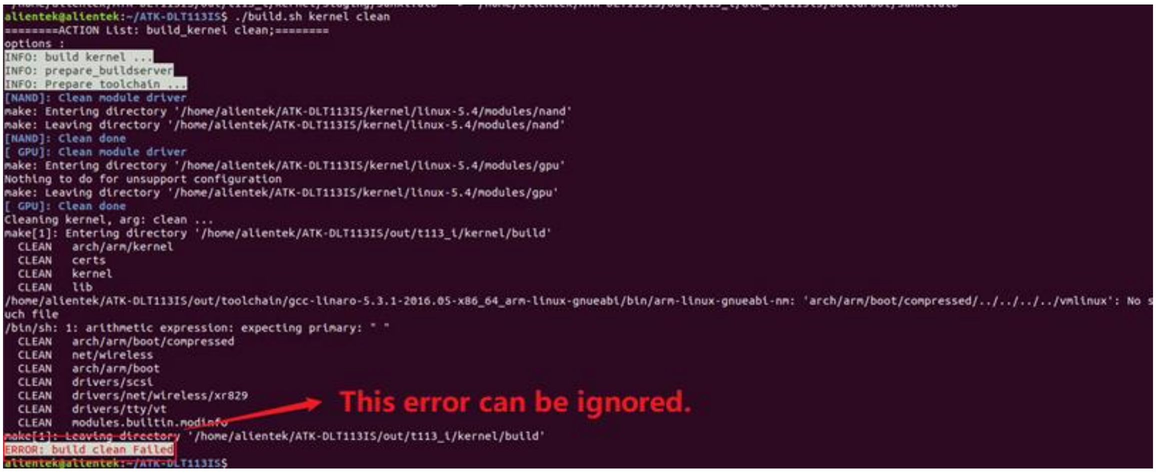
```
/home/alientek/ATK-DLT113IS/kernel/linux-5.4/bsp/modules/nand does not exist!
[ GPU]: Build module driver
make: Entering directory '/home/alientek/ATK-DLT113IS/kernel/linux-5.4/modules/gpu'
Nothing to do for unsupport configuration
make: Leaving directory '/home/alientek/ATK-DLT113IS/kernel/linux-5.4/modules/gpu'
make: Entering directory '/home/alientek/ATK-DLT113IS/kernel/linux-5.4/modules/gpu'
Nothing to do for unsupport configuration
make: Leaving directory '/home/alientek/ATK-DLT113IS/kernel/linux-5.4/modules/gpu'
[ GPU]: Build done
/home/alientek/ATK-DLT113IS/kernel/linux-5.4/bsp/modules/gpu does not exist!
---build dts for sun8iw20p1 atk_dlt113is-----
'/home/alientek/ATK-DLT113IS/out/t113_i/kernel/build/arch/arm/boot/dts//board.dtb' -> '/home/alientek/ATK-DLT113IS/out/t113_i/kernel/staging/sunxi.dtb'
Use init ramdisk file: '/home/alientek/ATK-DLT113IS/kernel/linux-5.4/rootfs_32bit.cpio.gz'.
old DTB_OFFSET: 14680064(0x00e00000)
new DTB_OFFSET: 18402048(0x0118cb00)
bootimg_build
Copy boot.img to output directory ...

sun8iw20p1 compile all(Kernel+modules+boot.img) successful


INFO: build dts ...
INFO: Prepare toolchain ...
removed '/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/.board.dtb.d.dtc.tmp'
removed '/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/.board.dtb.dts.tmp'
'/home/alientek/ATK-DLT113IS/out/t113_i/kernel/build/arch/arm/boot/dts/.board.dtb.d.dtc.tmp' -> '/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/.board.dtb.d.
dtc.tmp'
'/home/alientek/ATK-DLT113IS/out/t113_i/kernel/build/arch/arm/boot/dts/.board.dtb.dts.tmp' -> '/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/.board.dtb.dts.
tmp'
'/home/alientek/ATK-DLT113IS/out/t113_i/kernel/staging/sunxi.dtb' -> '/home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/openwrt/sunxi.dtb'
alientek@alientek:~/ATK-DLT113IS$
```

Figure 1.3-22 Kernel compilation completed

1.3.9 Kernel clearance

If the compilation of the kernel can be cleared, the following command can be executed (you can ignore the "ERROR" message):

```
./build.sh kernel clean
```



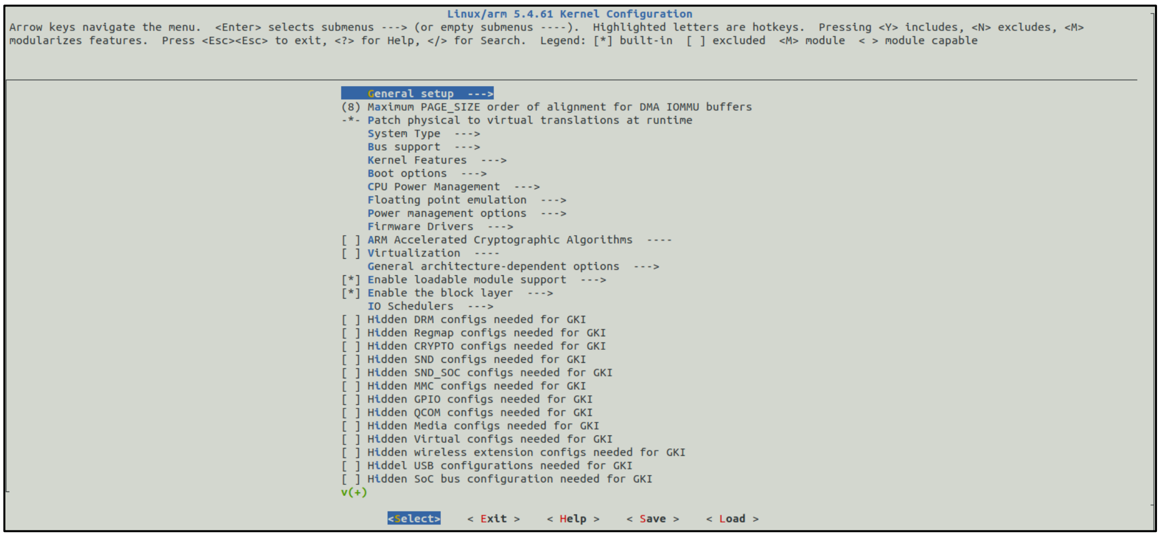
```
alientek@alientek:~/ATK-DLT113IS$ ./build.sh kernel clean
========ACTION List: build_kernel clean;========
options :
INFO: build kernel ...
INFO: prepare_buildserver
INFO: Prepare toolchain ...
[NAND]: Clean module driver
make: Entering directory '/home/alientek/ATK-DLT113IS/kernel/linux-5.4/modules/nand'
make: Leaving directory '/home/alientek/ATK-DLT113IS/kernel/linux-5.4/modules/nand'
[NAND]: Clean done
[ GPU]: Clean module driver
make: Entering directory '/home/alientek/ATK-DLT113IS/kernel/linux-5.4/modules/gpu'
Nothing to do for unsupport configuration
make: Leaving directory '/home/alientek/ATK-DLT113IS/kernel/linux-5.4/modules/gpu'
[ GPU]: Clean done
Cleaning kernel, arg: clean ...
make[1]: Entering directory '/home/alientek/ATK-DLT113IS/out/t113_i/kernel/build'
  CLEAN   arch/arm/kernel
  CLEAN   certs
  CLEAN   kernel
  CLEAN   lib
/home/alientek/ATK-DLT113IS/out/toolchain/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi/bin/arm-linux-gnueabi-nm: 'arch/arm/boot/compressed/../../../../vmlinux': No s
uch file
/bin/sh: 1: arithmetic expression: expecting primary: " "
  CLEAN   arch/arm/boot/compressed
  CLEAN   net/wireless
  CLEAN   arch/arm/boot
  CLEAN   drivers/scsi
  CLEAN   drivers/net/wireless/xr829
  CLEAN   drivers/tty/vt
  CLEAN   modules.builtin.modinfo
make[1]: Leaving directory '/home/alientek/ATK-DLT113IS/out/t113_i/kernel/build'
ERROR: build clean Failed
alientek@alientek:~/ATK-DLT113IS$
```

Figure 1.3-23 Kernel cleanup

1.3.10 Kernel configuration

The configuration command for the kernel is as follows:

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

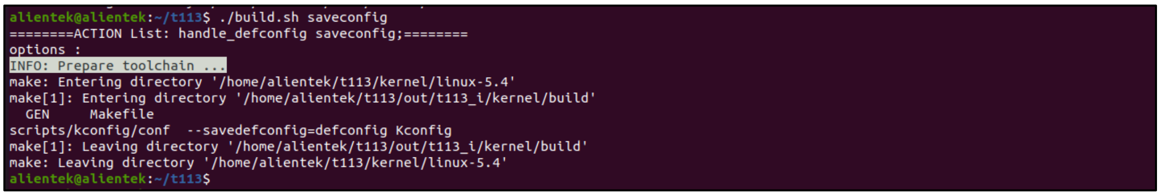


```
                        Linux/arm 5.4.61 Kernel Configuration
Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M>
modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                                                  General setup  --->
                                              (8) Maximum PAGE_SIZE order of alignment for DMA IOMMU buffers
                                              -*- Patch physical to virtual translations at runtime
                                                  System Type  --->
                                                  Bus support  --->
                                                  Kernel Features  --->
                                                  Boot options  --->
                                                  CPU Power Management  --->
                                                  Floating point emulation  --->
                                                  Power management options  --->
                                                  Firmware Drivers  --->
                                              [ ] ARM Accelerated Cryptographic Algorithms  ----
                                              [ ] Virtualization  ----
                                                  General architecture-dependent options  --->
                                              [*] Enable loadable module support  --->
                                              [*] Enable the block layer  --->
                                                  IO Schedulers  --->
                                              [ ] Hidden DRM configs needed for GKI
                                              [ ] Hidden Regmap configs needed for GKI
                                              [ ] Hidden CRYPTO configs needed for GKI
                                              [ ] Hidden SND configs needed for GKI
                                              [ ] Hidden SND_SOC configs needed for GKI
                                              [ ] Hidden MMC configs needed for GKI
                                              [ ] Hidden GPIO configs needed for GKI
                                              [ ] Hidden QCOM configs needed for GKI
                                              [ ] Hidden Media configs needed for GKI
                                              [ ] Hidden Virtual configs needed for GKI
                                              [ ] Hidden wireless extension configs needed for GKI
                                              [ ] Hiddel USB configurations needed for GKI
                                              [ ] Hidden SoC bus configuration needed for GKI
                                              v(+)

                                                      <Select>    < Exit >    < Help >    < Save >    < Load >
```

Figure 1.3-24 Kernel configuration diagram

After configuration is completed, you can run the following command to save the configuration file:

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



```
alientek@alientek:~/t113$ ./build.sh saveconfig
========ACTION List: handle_defconfig saveconfig;========
options :
INFO: Prepare toolchain ...
make: Entering directory '/home/alientek/t113/kernel/linux-5.4'
make[1]: Entering directory '/home/alientek/t113/out/t113_i/kernel/build'
  GEN     Makefile
scripts/kconfig/conf  --savedefconfig=defconfig Kconfig
make[1]: Leaving directory '/home/alientek/t113/out/t113_i/kernel/build'
make: Leaving directory '/home/alientek/t113/kernel/linux-5.4'
alientek@alientek:~/t113$
```

Figure 1.3-25 Save configuration file

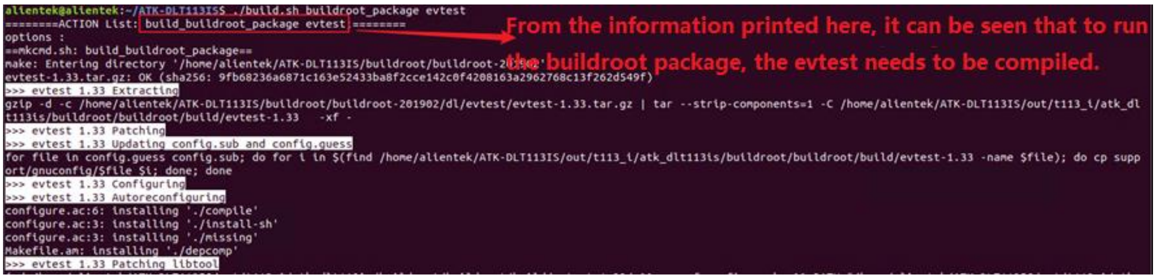
It will be directly saved to "device/config/chips/t113_i/configs/atk_dlt113is/linux-5.4/alientek_t113_kernel_defconfig". This file will be determined based on your configuration file.

1.3.11 Buildroot's package

If you need to compile the buildroot package separately, you can run the following command for testing. Let's test with the evtest command.

1. Compile

```
./build.sh buildroot_package evtest
```



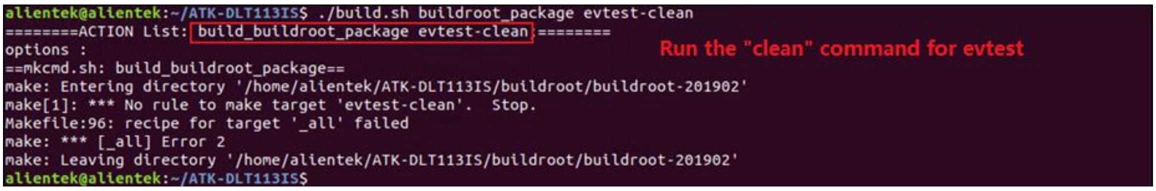
```
alientek@alientek:~/ATK-DLT113IS$ ./build.sh buildroot_package evtest
========ACTION List: build_buildroot_package evtest ========
options :
==mkcmd.sh: build_buildroot_package==
make: Entering directory '/home/alientek/ATK-DLT113IS/buildroot/buildroot-201902'
evtest-1.33.tar.gz: OK (sha256: 9fb68236a6871c163e52433ba8f2cce142c0f4208163a2962768c13f262d549f)
>>> evtest 1.33 Extracting
gzip -d -c /home/alientek/ATK-DLT113IS/buildroot/buildroot-201902/dl/evtest/evtest-1.33.tar.gz | tar --strip-components=1 -C /home/alientek/ATK-DLT113IS/out/t113_i/atk_dl
t113is/buildroot/buildroot/build/evtest-1.33   -xf -
>>> evtest 1.33 Patching
>>> evtest 1.33 Updating config.sub and config.guess
for file in config.guess config.sub; do for i in $(find /home/alientek/ATK-DLT113IS/out/t113_i/atk_dlt113is/buildroot/buildroot/build/evtest-1.33 -name $file); do cp supp
ort/gnuconfig/$file $i; done; done
>>> evtest 1.33 Configuring
>>> evtest 1.33 Autoreconfiguring
configure.ac:6: installing './compile'
configure.ac:3: installing './install-sh'
configure.ac:3: installing './missing'
Makefile.am: installing './depcomp'
>>> evtest 1.33 Patching libtool
```

From the information printed here, it can be seen that to run the buildroot package, the evtest needs to be compiled.

Figure 1.3-26 evtest compilation

2. Clearing

```
./build.sh buildroot_package evtest-clean
or
./build.sh buildroot_package evtest-dirclean
```



```
alientek@alientek:~/ATK-DLT113IS$ ./build.sh buildroot_package evtest-clean
========ACTION List: build_buildroot_package evtest-clean========
options :
==mkcmd.sh: build_buildroot_package==
make: Entering directory '/home/alientek/ATK-DLT113IS/buildroot/buildroot-201902'
make[1]: *** No rule to make target 'evtest-clean'.  Stop.
Makefile:96: recipe for target '_all' failed
make: *** [_all] Error 2
make: Leaving directory '/home/alientek/ATK-DLT113IS/buildroot/buildroot-201902'
alientek@alientek:~/ATK-DLT113IS$
```

Run the "clean" command for evtest

Figure 1.3-27 evtest clean test

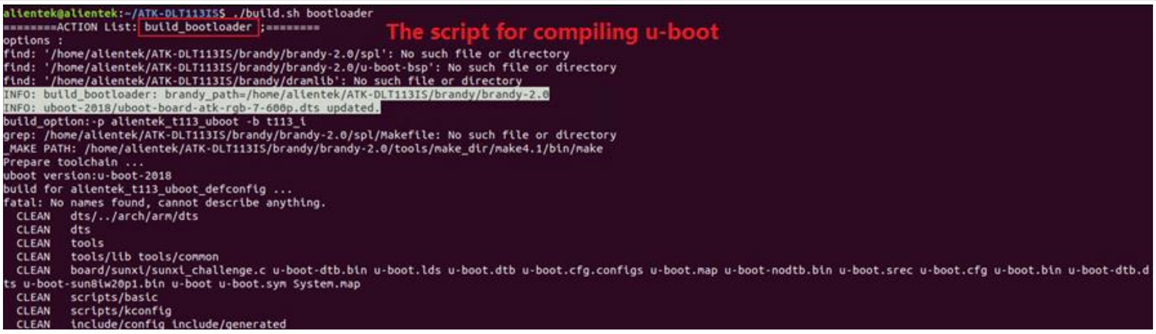
The author has listed all the commonly used command tests as examples. As shown in the table below:

Command / Objective	Explanation	Example
<pkg>	Compile and build the <pkg> package and its dependencies	./build.sh buildroot_package evtest
<pkg>-source	Download the source code to the directory	./build.sh buildroot_package evtest-source
<pkg>-extract	Extract the source code to the package build directory	./build.sh buildroot_package evtest-extract
<pkg>-patch	Apply patches to the package build directory	./build.sh buildroot_package evtest-patch
<pkg>-depends	Compile the dependencies of the <pkg> package	./build.sh buildroot_package evtest-depends
<pkg>-configure	Configure the pre-compilation commands (downloading, extraction, patching, and compiling dependencies)	./build.sh buildroot_package evtest-configure
<pkg>-build	Run the compilation command	make rkwifibt-build
<pkg>-show-depends	List the dependencies of the <pkg> package	./build.sh buildroot_package evtest-show-depends
<pkg>-show-rdepends	List the software packages that <pkg> package is a dependency of	./build.sh buildroot_package evtest-show-rdepends
<pkg>-graph-depends	Generate the dependencies of the <pkg> package in PDF format	make rkwifibt-graph-depends
<pkg>-graph-rdepends	Generate the software packages that <pkg> package is a dependency of in PDF format	./build.sh buildroot_package evtest-graph-rdepends
<pkg>-dirclean	Delete the entire software package build directory	./build.sh buildroot_package evtest-dirclean
<pkg>-reconfigure	Re-run the configuration command	./build.sh buildroot_package evtest-reconfigure
<pkg>-rebuild	Re-run the compilation command	./build.sh buildroot_package evtest-rebuild

1.3.12 Uboot Compilation

Uboot is only compiled but not cleared. The compilation command is as follows:

```
./build.sh bootloader
```



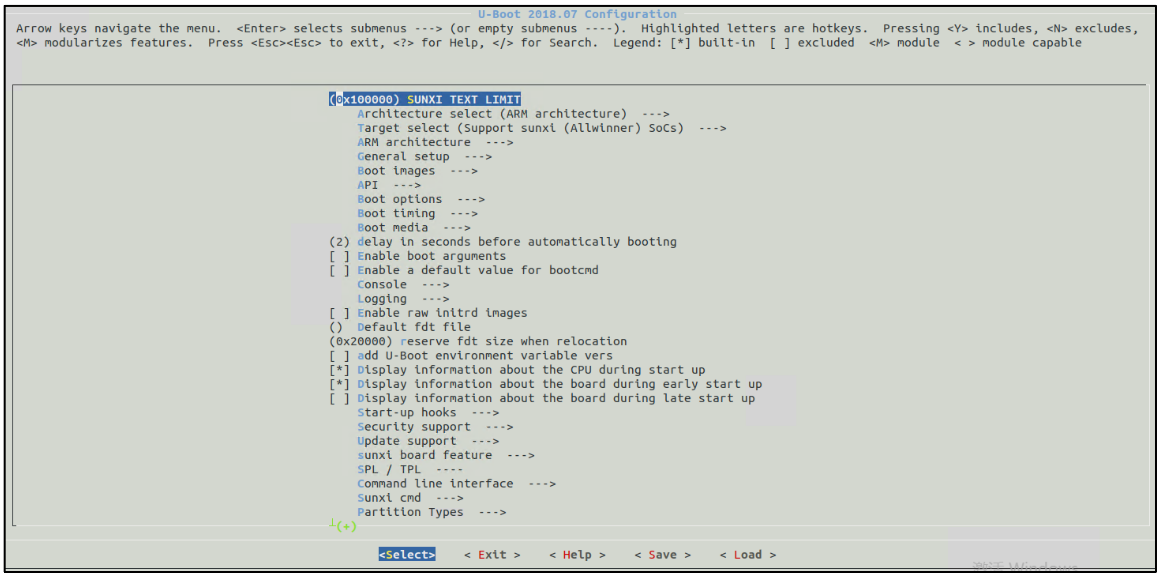
```
alientek@alientek:~/ATK-DLT113IS$ ./build.sh bootloader
========ACTION List: build_bootloader ;========
options :
find: '/home/alientek/ATK-DLT113IS/brandy/brandy-2.0/spl': No such file or directory
find: '/home/alientek/ATK-DLT113IS/brandy/brandy-2.0/u-boot-bsp': No such file or directory
find: '/home/alientek/ATK-DLT113IS/brandy/dramlib': No such file or directory
INFO: build_bootloader: brandy_path=/home/alientek/ATK-DLT113IS/brandy/brandy-2.0
INFO: uboot-2018/uboot-board-atk-rgb-7-600p.dts updated.
build_option:-p alientek_t113_uboot -b t113_i
grep: /home/alientek/ATK-DLT113IS/brandy/brandy-2.0/spl/Makefile: No such file or directory
_MAKE PATH: /home/alientek/ATK-DLT113IS/brandy/brandy-2.0/tools/make_dir/make4.1/bin/make
Prepare toolchain ...
uboot version:u-boot-2018
build for alientek_t113_uboot_defconfig ...
fatal: No names found, cannot describe anything.
  CLEAN   dts/../arch/arm/dts
  CLEAN   dts
  CLEAN   tools
  CLEAN   tools/lib tools/common
  CLEAN   board/sunxi/sunxi_challenge.c u-boot-dtb.bin u-boot.lds u-boot.dtb u-boot.cfg.configs u-boot.map u-boot-nodtb.bin u-boot.srec u-boot.cfg u-boot.bin u-boot-dtb.d
ts u-boot-sun8iw20p1.bin u-boot u-boot.sym System.map
  CLEAN   scripts/basic
  CLEAN   scripts/kconfig
  CLEAN   include/config include/generated
```

Figure 1.3-28 U-Boot compilation print information

1.3.13 U-Boot configuration

Use the following command to enable the defconfig configuration of U-Boot, as shown in the following command:

```
./build.sh uboot_menuconfig
```



```
                              U-Boot 2018.07 Configuration
Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
<M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

    (0x100000) SUNXI TEXT LIMIT
        Architecture select (ARM architecture)  --->
        Target select (Support sunxi (Allwinner) SoCs)  --->
        ARM architecture  --->
        General setup  --->
        Boot images  --->
        API  --->
        Boot options  --->
        Boot timing  --->
        Boot media  --->
    (2) delay in seconds before automatically booting
    [ ] Enable boot arguments
    [ ] Enable a default value for bootcmd
        Console  --->
        Logging  --->
    [ ] Enable raw initrd images
    ()  Default fdt file
    (0x20000) reserve fdt size when relocation
    [ ] add U-Boot environment variable vers
    [*] Display information about the CPU during start up
    [*] Display information about the board during early start up
    [ ] Display information about the board during late start up
        Start-up hooks  --->
        Security support  --->
        Update support  --->
        sunxi board feature  --->
        SPL / TPL  ----
        Command line interface  --->
        Sunxi cmd  --->
        Partition Types  --->
    ↓(+)

          <Select>    < Exit >    < Help >    < Save >    < Load >
```

Figure 1.3-29 The configuration diagram of Ubuntu

It will automatically call the configuration in the "out" directory. The modification is only for the output files and cannot directly modify the "alientek_t113_uboot_defconfig" file. Run the following command to save the corresponding file under "alientek_t113_uboot_defconfig".

```
./build.sh uboot_saveconfig
```

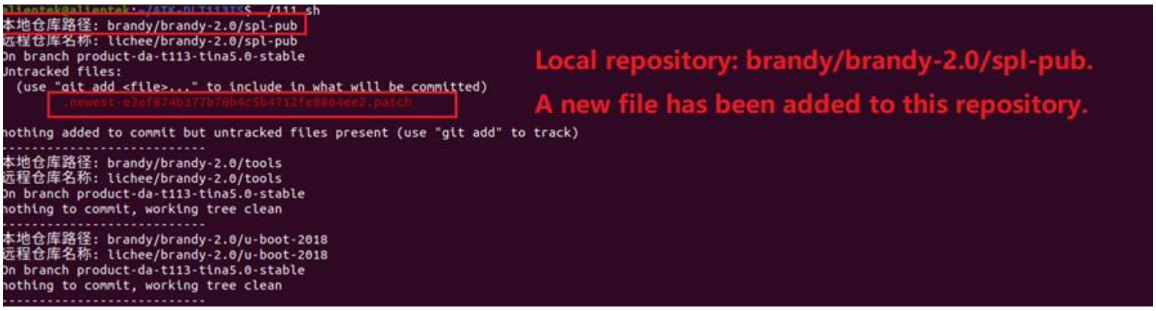
Chapter 2. SDK Usage Tips

2.1 List All Repositories

The SDK code of Allwinner is managed using repo, and it is divided into many git repositories. When we forget which repository a certain file was in and when it was modified, we can use the following script to check.

```
#!/bin/bash

.repo/repo/repo list | while read -r line; do
   # Extract the local path and the name of the remote repository
   local_path=$(echo $line | awk '{print $1}')
   remote_name=$(echo $line | awk '{print $3}')
      echo "本地仓库路径: $local_path"
   echo "远程仓库名称: $remote_name"
   git -C "$local_path" status
   echo "----------------------------"
done
```

Here, we use the "repo" command to list the paths of all repositories (including local and remote repositories). For each repository, we use the "git status" command to check the file modifications in the current repository. As shown in the figure below:



```
alientek@alientek:~/ATK-DLT113IS$ ./111.sh
本地仓库路径: brandy/brandy-2.0/spl-pub
远程仓库名称: lichee/brandy-2.0/spl-pub
On branch product-da-t113-tina5.0-stable
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .newest-e3ef874b377b78b4c5b4712fe8864ee2.patch

nothing added to commit but untracked files present (use "git add" to track)
----------------------------
本地仓库路径: brandy/brandy-2.0/tools
远程仓库名称: lichee/brandy-2.0/tools
On branch product-da-t113-tina5.0-stable
nothing to commit, working tree clean
----------------------------
本地仓库路径: brandy/brandy-2.0/u-boot-2018
远程仓库名称: lichee/brandy-2.0/u-boot-2018
On branch product-da-t113-tina5.0-stable
nothing to commit, working tree clean
----------------------------
```

Local repository: brandy/brandy-2.0/spl-pub.

A new file has been added to this repository.