DLNA Developer Guide

ID: RK-KF-YF-380

Release Version: V1.0.3

Release Date: 2021-11-09

Security Level: □Top-Secret □Secret □Internal ■Public

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian,PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

This document mainly introduces the development guide based on Buildroot DLNA

Product Version

Chipset	Kernel Version
RK3308	4.4

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Version	Author	Date	Change Description
V1.0.0	SCH	2018-05-28	Initial version
V1.0.1	Ruby Zhang	2019-05-29	Update the format of the document
V1.0.2	Ruby Zhang	2020-08-11	Update the company name; Convert the document to md format
V1.0.3	Ruby Zhang	2021-11-09	Update some expression

Contents

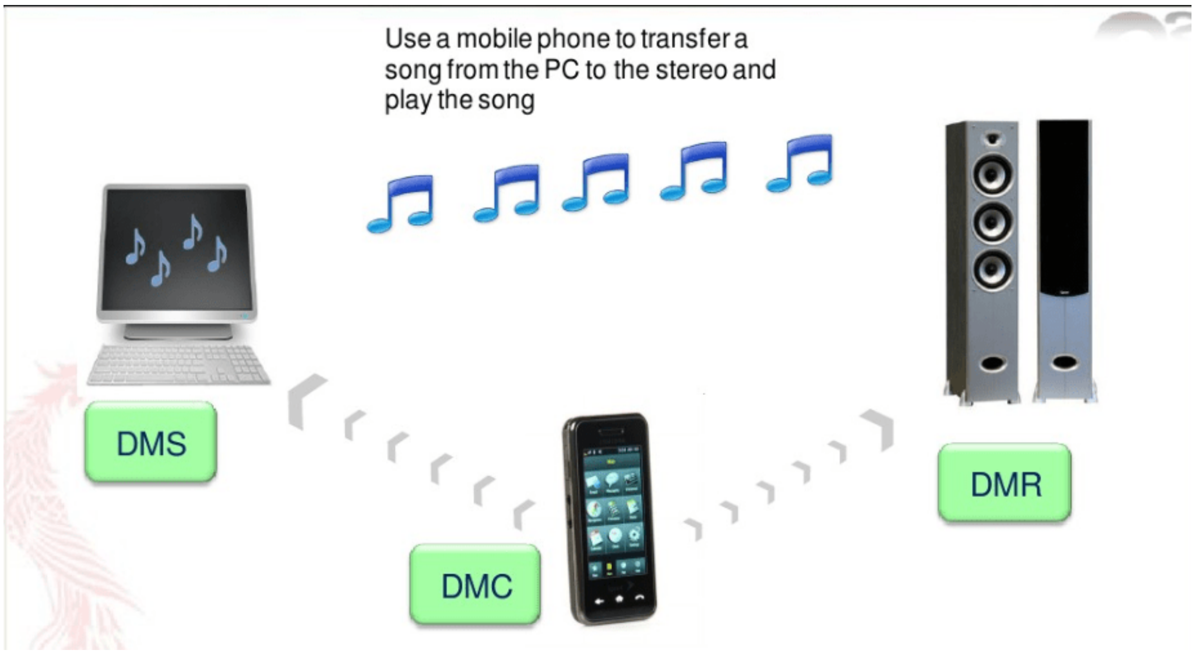
1. DLNA Overview

The full name of DLNA is DIGITAL LIVING NETWORK ALLIANCE (Digital Living Network Alliance).

DLNA was established on June 24, 2003, with the predecessor of DHWG (Digital Home Working Group). It was set up by Sony, Intel, Microsoft, etc., to solve the interconnection and intercommunication of wireless networks and wired networks of personal PCs, consumer electronics, and mobile devices, and makes it possible to unlimited sharing and growth of digital media and content services. The slogan of DLNA is Enjoy your music, photos and videos, anywhere anytime.

DLNA declared that the organization breakup officially on January 15, 2017, and will not update the DLNA standard in the future.

DLNA defines its application as 5 functional components. From bottom to top, they are: network interconnection, network protocol, media transmission, device discovery control and management, media format, the following is an application scenario of DLNA (the actual application scenario is far more than these)


Use a mobile phone to transfer a song from the PC to the stereo and play the song
DMS
DMC
DMR

2. Key Concepts of DLNA

- Home NetWork Device (HND), it refers to home device, which has a relatively large size and more comprehensive functions, and is mainly distinguished from mobile device. This type of device including following five types:

1. Digital Media Server (DMS), it provides media obtaining, recording, storage and output functions. At the same time, the content protection function is a mandatory requirement for DMS. DMS always includes the functions of DMP, and can include other intelligent functions, including device/user service management; rich user interface; media management/collection and distribution functions. Such as PCs, digital set-top boxes (with networking and storage functions), cameras, and so on.
2. DMP. Digital media player, it can search and obtain media content from DMS/M-DMS and play and render the display. Such as smart TV, home theater, etc.

3. DMC. Digital media controller, it used to search for the content of DMS and establishes the connection between DMS and DMR and controls the playback of the media. Such as remote control.
4. DMR. Digital media rendering device. After being configured by other devices, it can play the content from DMS. The difference with DMP is that DMR only has the function of obtaining media and playing, but without the function of finding and browsing media. Such as monitors, speakers, etc.
5. DMPr. Digital media printers provide printing services. Network printers and integrated printers belong to DMPr.

- Mobile Handheld Devices (MHD) handheld devices. Compared with home devices, the functions of handheld devices are relatively simplified, and the supported media formats are also different.

1. M-DMS. Similar to DMS, such as mobile phones, portable music players, etc.
2. M-DMP. Similar to DMP. Such as smart mobile TV.
3. M-DMD. Mobile multimedia download devices. Such as portable music player, car music player and smart electronic photo frame, etc.
4. M-DMU. Mobile multimedia download devices. Such as camera devices and mobile phones.
5. M-DMC. Similar to DMC. Such as DA, smart remote control. Mobile handheld devices do not define M-DMR, because mobile handheld devices pay attention to convenience and add search control functions, or they are just ordinary mobile TVs or radios.

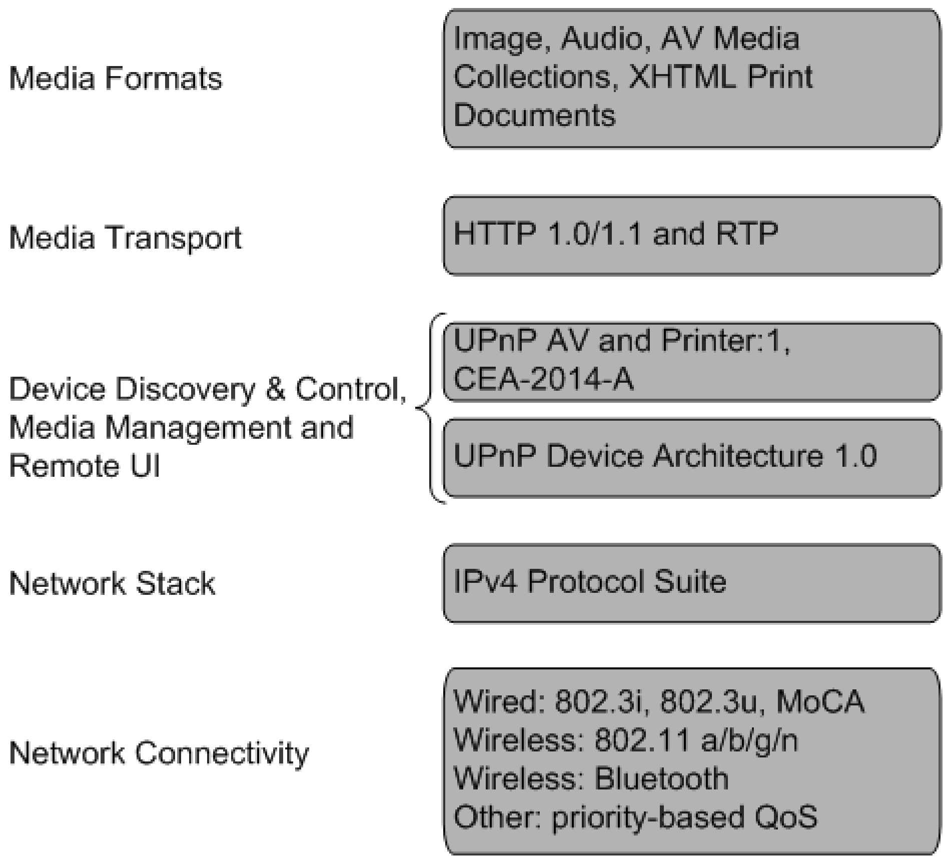
- Networked Infrastructure Devices (NID), networking support devices.

1. Mobile Network Connectivity Function (M-NCF), it provide physical media for various devices to access the mobile network. The purpose of DLNA is full wireless.
2. Interoperability Unit (MIU), media interaction devices, It provide media format conversion to support various device requirements.

3. DLNA Architecture

The DLNA architecture is an interconnected system, so it is logically similar to the OSI (Open System Interconnection) seven-layer network model.

The DLNA architecture is divided into 7 levels as shown in the figure below:


Media Formats
Image, Audio, AV Media Collections, XHTML Print Documents
Media Transport
HTTP 1.0/1.1 and RTP
Device Discovery & Control, Media Management and Remote UI
UPnP AV and Printer:1, CEA-2014-A
UPnP Device Architecture 1.0
Network Stack
IPv4 Protocol Suite
Network Connectivity
Wired: 802.3i, 802.3u, MoCA
Wireless: 802.11 a/b/g/n
Wireless: Bluetooth
Other: priority-based QoS

1. NetWorking Connectivity. Network interconnection ways, include physical connection standards, there are wired, such as Ethernet that meets the IEEE802.3 standard; and wireless, such as WiFi that meets the IEEE802.11a/g standard, which can reach 54Mbps, Bluetooth (802.15)) and other technologies are all mature technologies. Now OFDM and MIMO (802.11n) can reach 300Mbps, which has far exceeded the more popular 100Mbps Ethernet, but the product is not popular yet, and it will definitely be used in the future.
2. NetWorking Stack. Network protocol stack, DLNA interconnection transmission is based on the IPV4 protocol cluster. Either TCP or UDP can be used for transmission. This layer is similar to the OSI network layer.
3. Device Discovery&Control. This level is more essential and is the basic protocol architecture of DLNA. DLNA uses UPnP protocol to implement the function of device discovery and control. For this part, please refer to the documents from http://upnp.org/sdcps-and-certification/standards/device-architecture-documents/.
4. Media Management. Media management includes identification, management, distribution, and recording (saving) of media. UPnP AV Architecture:1 and UPnP Printer Architecture:1, these two UPnP documents will explain how to perform media management.
5. Media Transport: This layer uses HTTP (HyperText Transfer Protocol). It is the media transfer protocol that we usually use online. HTTP uses TCP for reliable transmission, and there is also a mixed UDP method of HTTP. The latest version of HTTP is HTTP 1.1. The optional protocol is RTP.
6. Media Formats. The format is similar to the encoding format Codec here. The encoding formats we usually talk about, such as Mpeg-2, AVC, and x264, are video encoding formats; PCM, mp3 (MPEG-2 Layer 3), aac, and flac are audio encoding formats. And avi, rmvb, mkv are media packaging formats, including video and audio and possibly subtitle streams. For example, a common file with a suffix of mkv, its video codec is x264, audio is aac, and its video and audio encoding belongs to the Mpeg-4 Codec Family.

4. Development Guide

We have a preliminary understanding of DLNA in the first 3 chapters. Next, we will build a DLNA environment to implement M-DMS (mobile phone QQ music player) to push music to DMR (RK3308 smart speaker).

We choose gmrender-resurrect open source code to build the DMR role, and find a mobile phone to install the QQ player to act as the M-DMS role.

4.1 Compilation

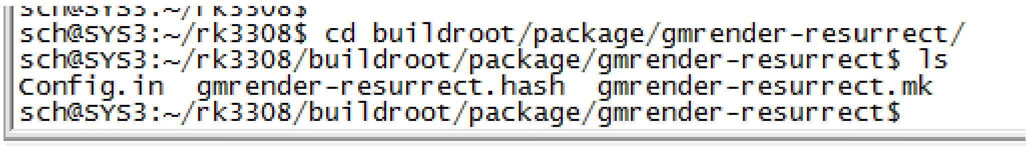
4.1.1 Version Confirmation

Before compiling, firstly, confirm the version of gmrender-resurrect and the relevant library libupnp to ensure that the version meets the following requirements:

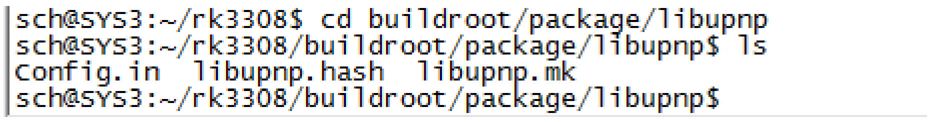
gmrender-resurrect version: 33600ab663f181c4f4f5c48aba25bf961760a300

Libupnp version: 1.6.21

The configuration information of the Buildroot package is in the corresponding folder under the Buildroot/package file, each folder contains 3 files, namely *.in, *.hash, *.mk, the following is the configuration of gmrender-resurrect and libupnp file screenshot:



```
sch@SYS3:~/rk3308$ cd buildroot/package/gmrender-resurrect/
sch@SYS3:~/rk3308/buildroot/package/gmrender-resurrect$ ls
Config.in  gmrender-resurrect.hash  gmrender-resurrect.mk
sch@SYS3:~/rk3308/buildroot/package/gmrender-resurrect$
```

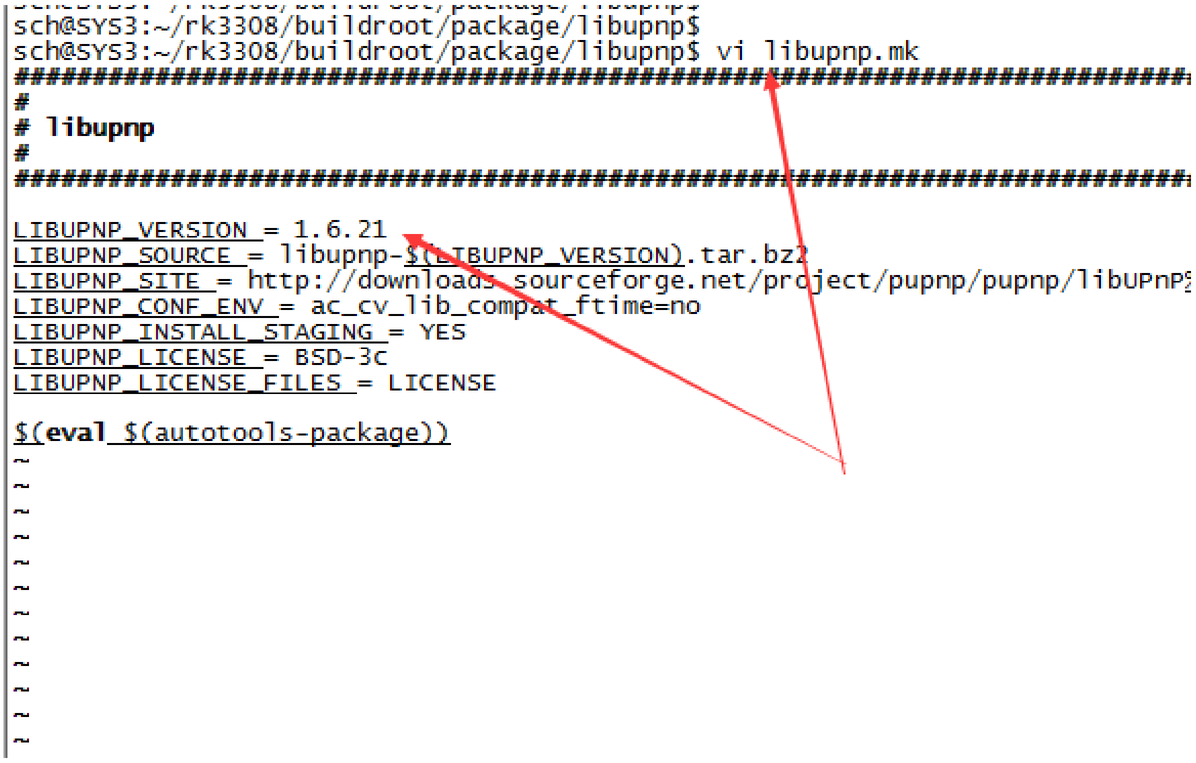


```
sch@SYS3:~/rk3308$ cd buildroot/package/libupnp
sch@SYS3:~/rk3308/buildroot/package/libupnp$ ls
Config.in  libupnp.hash  libupnp.mk
sch@SYS3:~/rk3308/buildroot/package/libupnp$
```

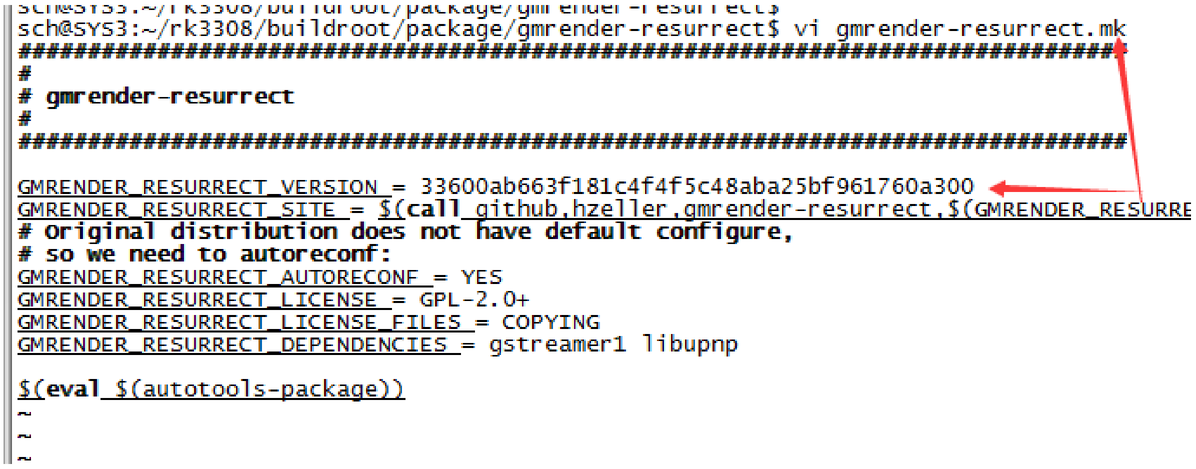
Note:

*.in file records the switch macro in makeconfig
*.hash file records the HASH value and version information of the code compression package
*.mk file records code compilation information and version information

We use the vi command to open the corresponding file to confirm the version information, as shown below:



```
sch@SYS3:~/rk3308/buildroot/package/libupnp$
sch@SYS3:~/rk3308/buildroot/package/libupnp$
sch@SYS3:~/rk3308/buildroot/package/libupnp$ vi libupnp.mk
################################################################################
#
# libupnp
#
################################################################################

LIBUPNP_VERSION = 1.6.21
LIBUPNP_SOURCE = libupnp-$(LIBUPNP_VERSION).tar.bz2
LIBUPNP_SITE = http://downloads.sourceforge.net/project/pupnp/pupnp/libUPnP
LIBUPNP_CONF_ENV = ac_cv_lib_compat_ftime=no
LIBUPNP_INSTALL_STAGING = YES
LIBUPNP_LICENSE = BSD-3c
LIBUPNP_LICENSE_FILES = LICENSE

$(eval $(autotools-package))
~
~
~
~
~
~
~
~
~
~
~
~
```

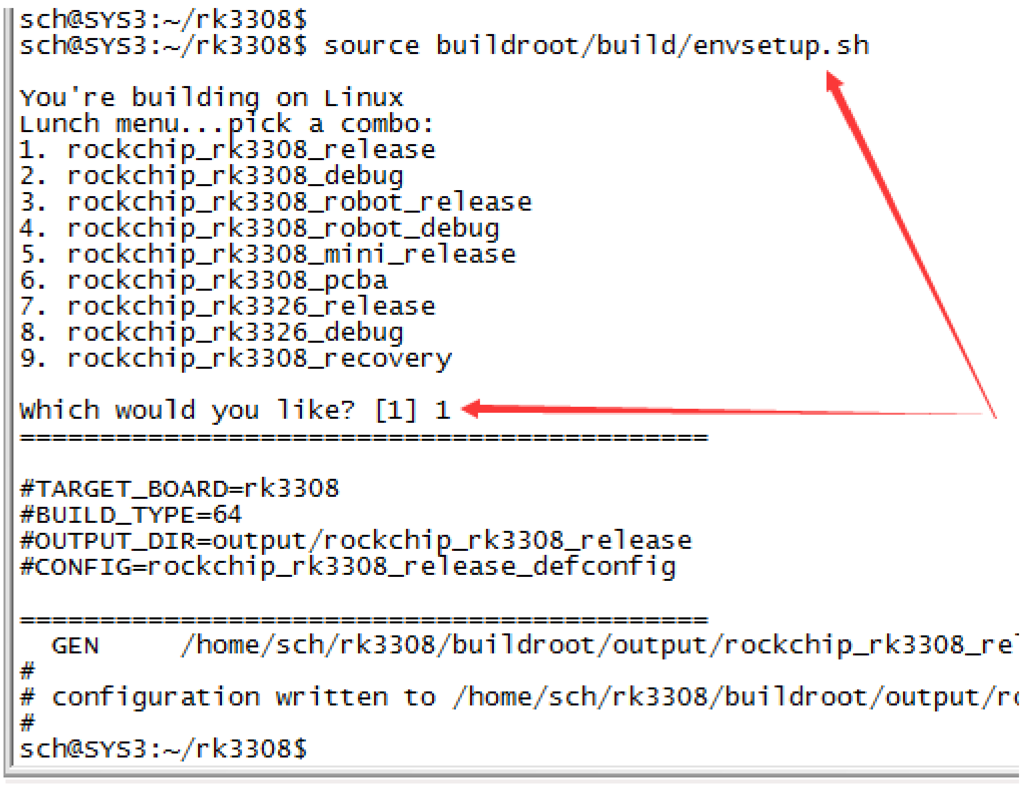


```
sch@SYS3:~/rk3308/buildroot/package/gmrender-resurrect$
sch@SYS3:~/rk3308/buildroot/package/gmrender-resurrect$ vi gmrender-resurrect.mk
################################################################################
#
# gmrender-resurrect
#
################################################################################

GMRENDER_RESURRECT_VERSION = 33600ab663f181c4f4f5c48aba25bf961760a300
GMRENDER_RESURRECT_SITE = $(call github,hzeller,gmrender-resurrect,$(GMRENDER_RESURRE
# Original distribution does not have default configure,
# so we need to autoreconf:
GMRENDER_RESURRECT_AUTORECONF = YES
GMRENDER_RESURRECT_LICENSE = GPL-2.0+
GMRENDER_RESURRECT_LICENSE_FILES = COPYING
GMRENDER_RESURRECT_DEPENDENCIES = gstreamer1 libupnp

$(eval $(autotools-package))
~
~
~
```

4.1.2 Configuration

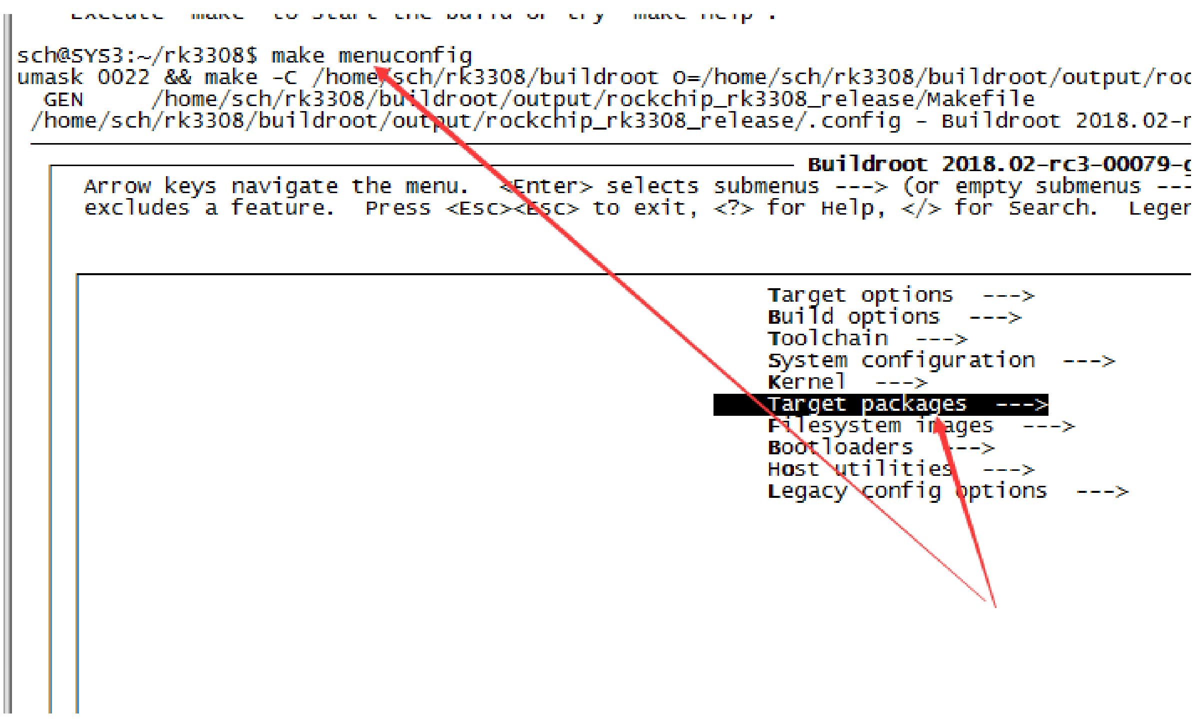
Follow the steps below to configure before compiling:

1. Set environment variables, use source buildroot/build/envsetup.sh

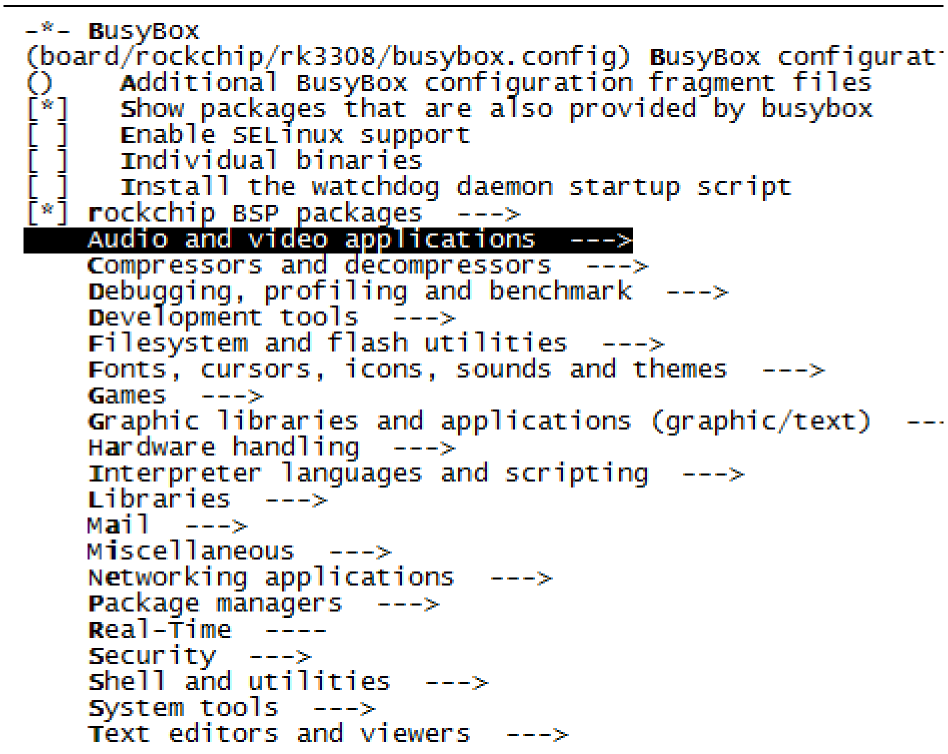


```
sch@SYS3:~/rk3308$
sch@SYS3:~/rk3308$ source buildroot/build/envsetup.sh

You're building on Linux
Lunch menu...pick a combo:
1. rockchip_rk3308_release
2. rockchip_rk3308_debug
3. rockchip_rk3308_robot_release
4. rockchip_rk3308_robot_debug
5. rockchip_rk3308_mini_release
6. rockchip_rk3308_pcba
7. rockchip_rk3326_release
8. rockchip_rk3326_debug
9. rockchip_rk3308_recovery

Which would you like? [1] 1
===========================================

#TARGET_BOARD=rk3308
#BUILD_TYPE=64
#OUTPUT_DIR=output/rockchip_rk3308_release
#CONFIG=rockchip_rk3308_release_defconfig

===========================================
  GEN     /home/sch/rk3308/buildroot/output/rockchip_rk3308_re
#
# configuration written to /home/sch/rk3308/buildroot/output/r
#
sch@SYS3:~/rk3308$
```

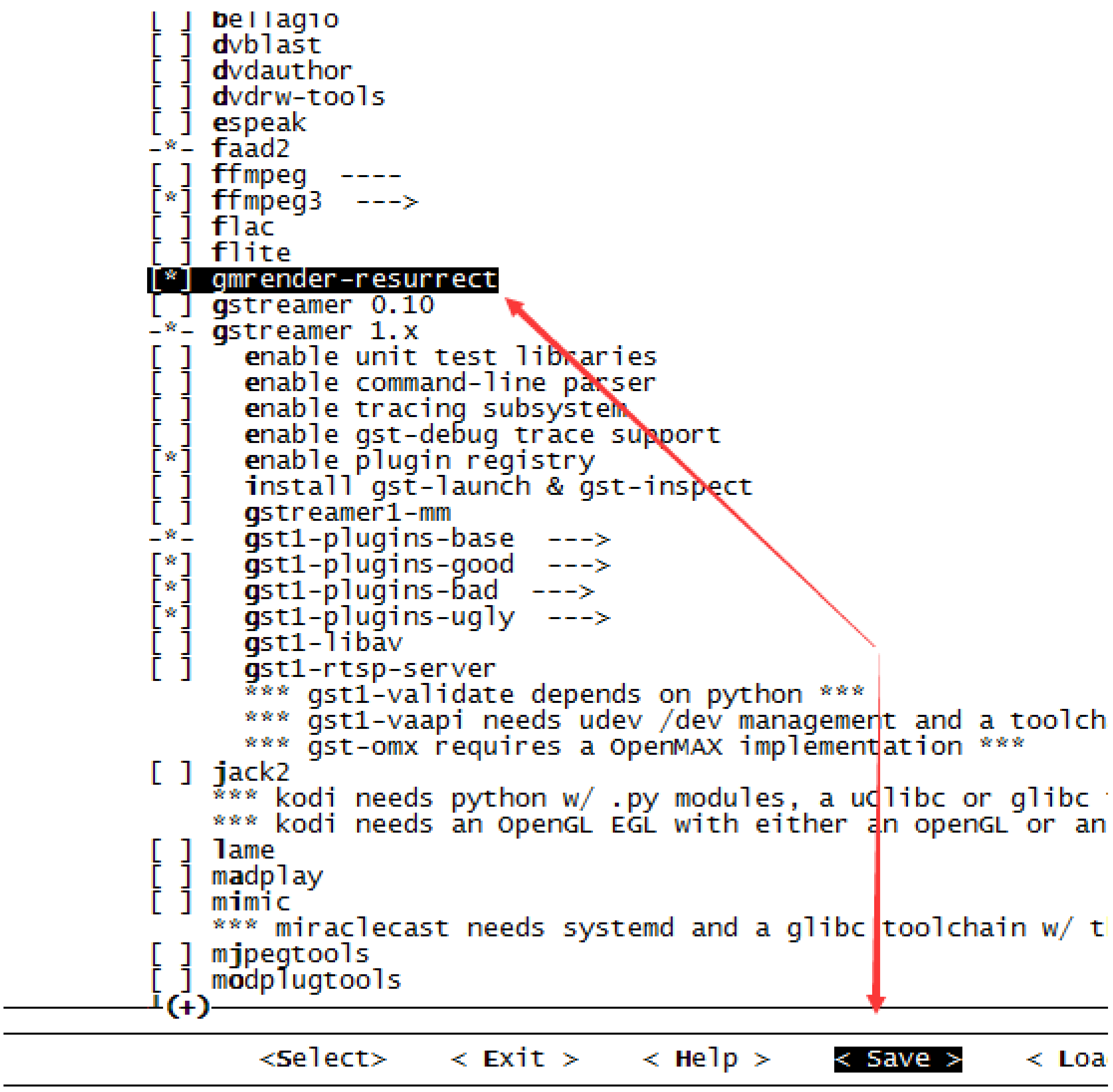
2. Configure menuconfig



```
sch@SYS3:~/rk3308$ make menuconfig
umask 0022 && make -C /home/sch/rk3308/buildroot O=/home/sch/rk3308/buildroot/output/roc
  GEN     /home/sch/rk3308/buildroot/output/rockchip_rk3308_release/Makefile
 /home/sch/rk3308/buildroot/output/rockchip_rk3308_release/.config - Buildroot 2018.02-r

                                                 Buildroot 2018.02-rc3-00079-g
    Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ---
    excludes a feature.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Leger

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

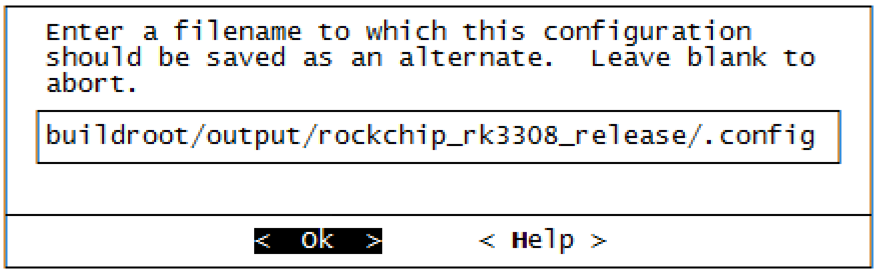


```
 -*- BusyBox
 (board/rockchip/rk3308/busybox.config) BusyBox configurat
 ()    Additional BusyBox configuration fragment files
 [*]   Show packages that are also provided by busybox
 [ ]   Enable SELinux support
 [ ]   Individual binaries
 [ ]   Install the watchdog daemon startup script
 [*] rockchip BSP packages  --->
     Audio and video applications  --->
     Compressors and decompressors  --->
     Debugging, profiling and benchmark  --->
     Development tools  --->
     Filesystem and flash utilities  --->
     Fonts, cursors, icons, sounds and themes  --->
     Games  --->
     Graphic libraries and applications (graphic/text)  --
     Hardware handling  --->
     Interpreter languages and scripting  --->
     Libraries  --->
     Mail  --->
     Miscellaneous  --->
     Networking applications  --->
     Package managers  --->
     Real-Time  ----
     Security  --->
     Shell and utilities  --->
     System tools  --->
     Text editors and viewers  --->
```



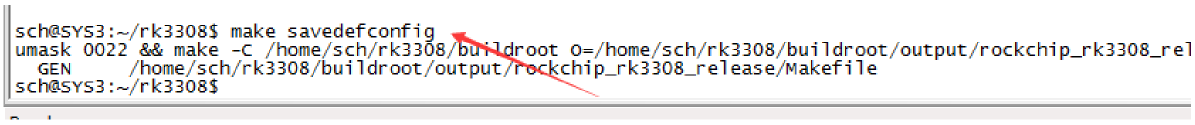
```
[ ] bellagio
[ ] dvblast
[ ] dvdauthor
[ ] dvdrw-tools
[ ] espeak
-*- faad2
[ ] ffmpeg  ----
[*] ffmpeg3  --->
[ ] flac
[ ] flite
[*] gmrender-resurrect
[ ] gstreamer 0.10
-*- gstreamer 1.x
[ ]   enable unit test libraries
[ ]   enable command-line parser
[ ]   enable tracing subsystem
[ ]   enable gst-debug trace support
[*]   enable plugin registry
[ ]   install gst-launch & gst-inspect
[ ]   gstreamer1-mm
-*-   gst1-plugins-base  --->
[*]   gst1-plugins-good  --->
[*]   gst1-plugins-bad  --->
[*]   gst1-plugins-ugly  --->
[ ]   gst1-libav
[ ]   gst1-rtsp-server
      *** gst1-validate depends on python ***
      *** gst1-vaapi needs udev /dev management and a toolch
      *** gst-omx requires a OpenMAX implementation ***
[ ] jack2
    *** kodi needs python w/ .py modules, a uclibc or glibc
    *** kodi needs an OpenGL EGL with either an openGL or an
[ ] lame
[ ] madplay
[ ] mimic
    *** miraclecast needs systemd and a glibc toolchain w/ t
[ ] mjpegtools
[ ] modplugtools
 (+)

        <Select>    < Exit >    < Help >    < Save >    < Loa
```

If gmrender-resurrect is not selected (the picture is selected), press Y to select and save:



```
Enter a filename to which this configuration
should be saved as an alternate.  Leave blank to
abort.

buildroot/output/rockchip_rk3308_release/.config

        < Ok >        < Help >
```

If it cannot compile after saving the config, you need to use make savedefconfig to export the configuration:

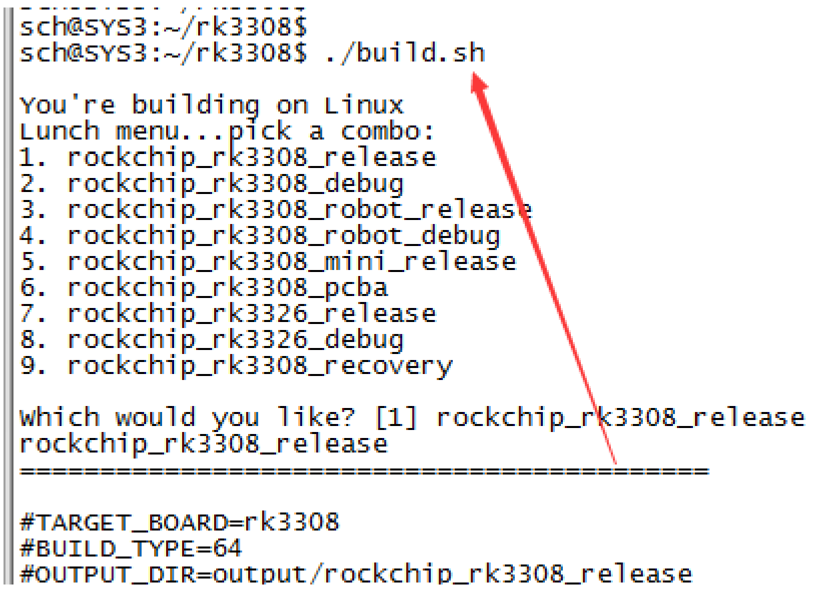


```
sch@SYS3:~/rk3308$ make savedefconfig
umask 0022 && make -C /home/sch/rk3308/buildroot O=/home/sch/rk3308/buildroot/output/rockchip_rk3308_rel
  GEN     /home/sch/rk3308/buildroot/output/rockchip_rk3308_release/Makefile
sch@SYS3:~/rk3308$
```

After successfully exporting the configuration, we can compile it, please refer to chapter 4.1.3

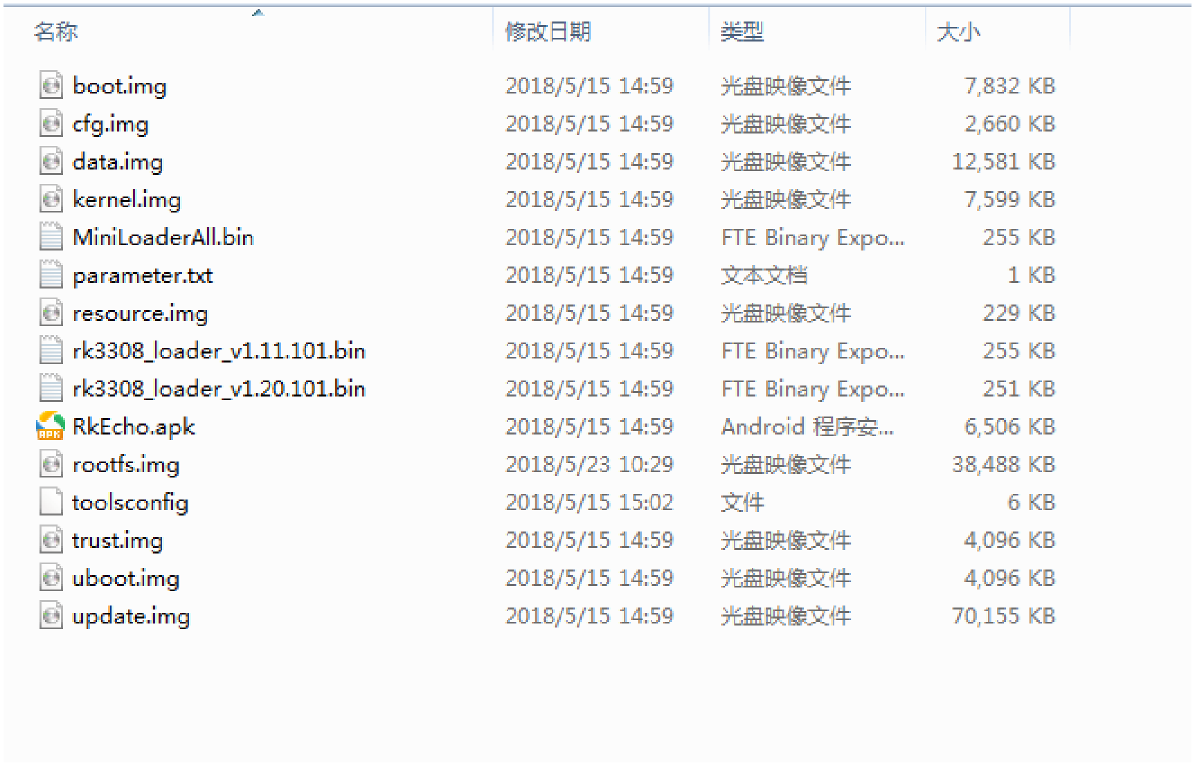
4.1.3 Compilation

1. If the compilation fails, please use build.sh for full compilation



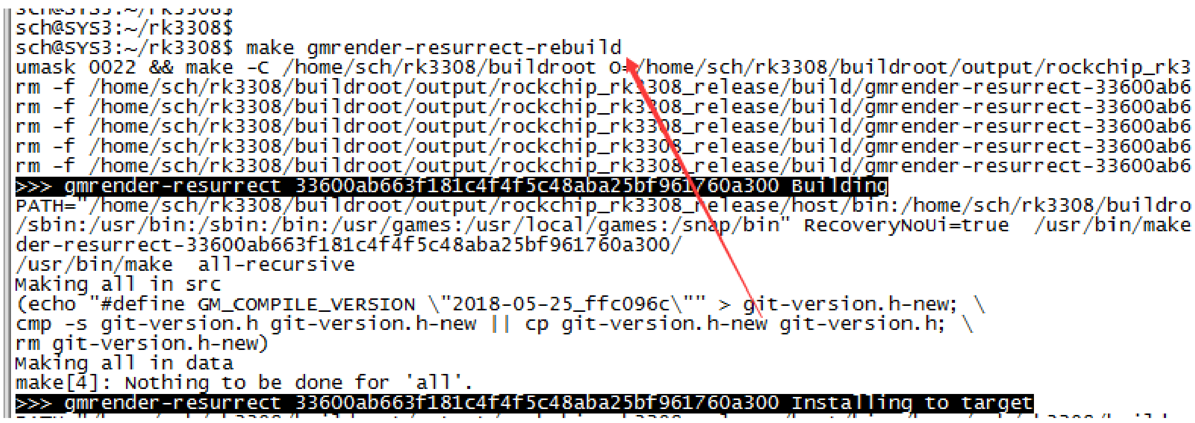
```
sch@SYS3:~/rk3308$
sch@SYS3:~/rk3308$ ./build.sh

You're building on Linux
Lunch menu...pick a combo:
1. rockchip_rk3308_release
2. rockchip_rk3308_debug
3. rockchip_rk3308_robot_release
4. rockchip_rk3308_robot_debug
5. rockchip_rk3308_mini_release
6. rockchip_rk3308_pcba
7. rockchip_rk3326_release
8. rockchip_rk3326_debug
9. rockchip_rk3308_recovery

Which would you like? [1] rockchip_rk3308_release
rockchip_rk3308_release
===========================================

#TARGET_BOARD=rk3308
#BUILD_TYPE=64
#OUTPUT_DIR=output/rockchip_rk3308_release
```

After full compilation, a subdirectory IMAGE\RK3308-EVB-DMIC-I2S-V10_20180515.1456_RELEASE_TEST\IMAGES will be generated under the IMAGE directory, which contains all the IMG required by RK3308, as shown in the following figure:



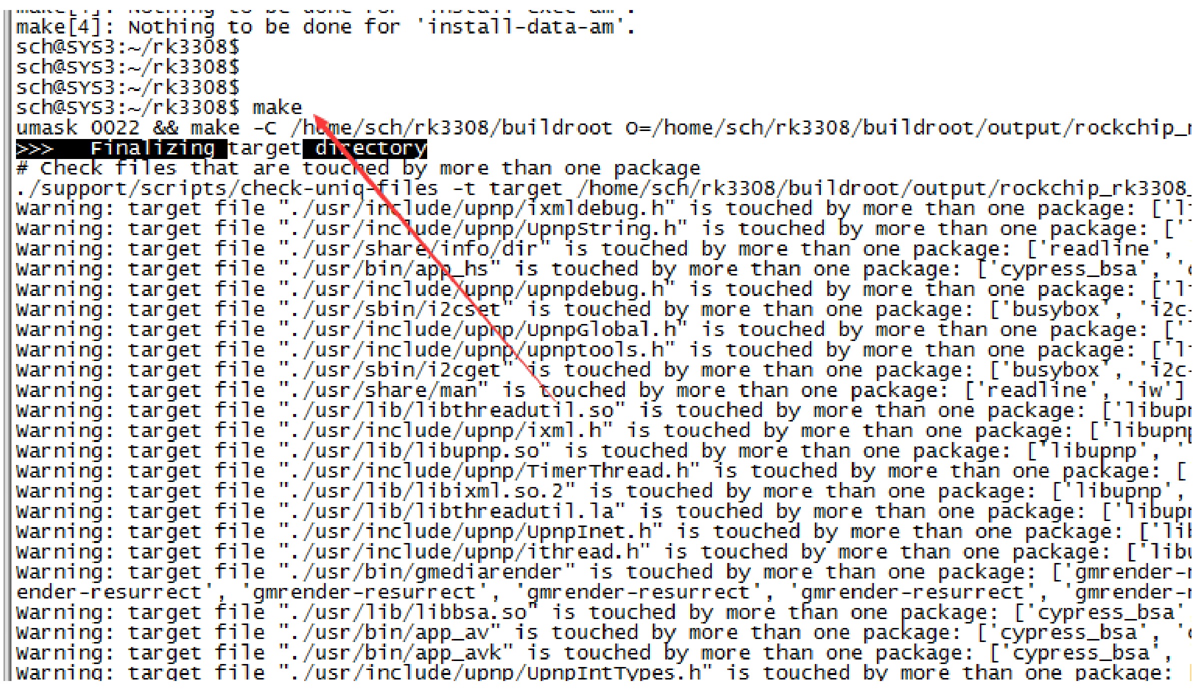
名称	修改日期	类型	大小
boot.img	2018/5/15 14:59	光盘映像文件	7,832 KB
cfg.img	2018/5/15 14:59	光盘映像文件	2,660 KB
data.img	2018/5/15 14:59	光盘映像文件	12,581 KB
kernel.img	2018/5/15 14:59	光盘映像文件	7,599 KB
MiniLoaderAll.bin	2018/5/15 14:59	FTE Binary Expo...	255 KB
parameter.txt	2018/5/15 14:59	文本文档	1 KB
resource.img	2018/5/15 14:59	光盘映像文件	229 KB
rk3308_loader_v1.11.101.bin	2018/5/15 14:59	FTE Binary Expo...	255 KB
rk3308_loader_v1.20.101.bin	2018/5/15 14:59	FTE Binary Expo...	251 KB
RkEcho.apk	2018/5/15 14:59	Android 程序安...	6,506 KB
rootfs.img	2018/5/23 10:29	光盘映像文件	38,488 KB
toolsconfig	2018/5/15 15:02	文件	6 KB
trust.img	2018/5/15 14:59	光盘映像文件	4,096 KB
uboot.img	2018/5/15 14:59	光盘映像文件	4,096 KB
update.img	2018/5/15 14:59	光盘映像文件	70,155 KB

If the system has been fully compiled, we can use make gmrender-resurrect-rebuild to compile only gmrender-resurrect and its related libraries.



```
sch@SYS3:~/rk3308$
sch@SYS3:~/rk3308$ make gmrender-resurrect-rebuild
umask 0022 && make -C /home/sch/rk3308/buildroot O=/home/sch/rk3308/buildroot/output/rockchip_rk3
rm -f /home/sch/rk3308/buildroot/output/rockchip_rk3308_release/build/gmrender-resurrect-33600ab6
rm -f /home/sch/rk3308/buildroot/output/rockchip_rk3308_release/build/gmrender-resurrect-33600ab6
rm -f /home/sch/rk3308/buildroot/output/rockchip_rk3308_release/build/gmrender-resurrect-33600ab6
rm -f /home/sch/rk3308/buildroot/output/rockchip_rk3308_release/build/gmrender-resurrect-33600ab6
rm -f /home/sch/rk3308/buildroot/output/rockchip_rk3308_release/build/gmrender-resurrect-33600ab6
>>> gmrender-resurrect 33600ab663f181c4f4f5c48aba25bf961760a300 Building
PATH="/home/sch/rk3308/buildroot/output/rockchip_rk3308_release/host/bin:/home/sch/rk3308/buildro
/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin" RecoveryNoUi=true  /usr/bin/make
der-resurrect-33600ab663f181c4f4f5c48aba25bf961760a300/
/usr/bin/make  all-recursive
Making all in src
(echo "#define GM_COMPILE_VERSION \"2018-05-25_ffc096c\"" > git-version.h-new; \
cmp -s git-version.h git-version.h-new || cp git-version.h-new git-version.h; \
rm git-version.h-new)
Making all in data
make[4]: Nothing to be done for 'all'.
>>> gmrender-resurrect 33600ab663f181c4f4f5c48aba25bf961760a300 Installing to target
```

After gmrender-resurrect is compiled separately, you need to use the make command to generate rootfs.



```
make[4]: Nothing to be done for 'install-data-am'.
sch@SYS3:~/rk3308$
sch@SYS3:~/rk3308$
sch@SYS3:~/rk3308$
sch@SYS3:~/rk3308$ make
umask 0022 && make -C /home/sch/rk3308/buildroot O=/home/sch/rk3308/buildroot/output/rockchip_r
>>>   Finalizing target directory
# Check files that are touched by more than one package
./support/scripts/check-uniq-files -t target /home/sch/rk3308/buildroot/output/rockchip_rk3308_
Warning: target file "./usr/include/upnp/ixmldebug.h" is touched by more than one package: ['l
Warning: target file "./usr/include/upnp/UpnpString.h" is touched by more than one package: ['
Warning: target file "./usr/share/info/dir" is touched by more than one package: ['readline',
Warning: target file "./usr/bin/app_hs" is touched by more than one package: ['cypress_bsa', '
Warning: target file "./usr/include/upnp/upnpdebug.h" is touched by more than one package: ['l
Warning: target file "./usr/sbin/i2cset" is touched by more than one package: ['busybox', 'i2c-
Warning: target file "./usr/include/upnp/UpnpGlobal.h" is touched by more than one package: ['
Warning: target file "./usr/include/upnp/upnptools.h" is touched by more than one package: ['l
Warning: target file "./usr/sbin/i2cget" is touched by more than one package: ['busybox', 'i2c-
Warning: target file "./usr/share/man" is touched by more than one package: ['readline', 'iw']
Warning: target file "./usr/lib/libthreadutil.so" is touched by more than one package: ['libupn
Warning: target file "./usr/include/upnp/ixml.h" is touched by more than one package: ['libupnp
Warning: target file "./usr/lib/libupnp.so" is touched by more than one package: ['libupnp', '
Warning: target file "./usr/include/upnp/TimerThread.h" is touched by more than one package: [
Warning: target file "./usr/lib/libixml.so.2" is touched by more than one package: ['libupnp',
Warning: target file "./usr/lib/libthreadutil.la" is touched by more than one package: ['libupn
Warning: target file "./usr/include/upnp/UpnpInet.h" is touched by more than one package: ['lib
Warning: target file "./usr/include/upnp/ithread.h" is touched by more than one package: ['libu
Warning: target file "./usr/bin/gmediarender" is touched by more than one package: ['gmrender-r
ender-resurrect', 'gmrender-resurrect', 'gmrender-resurrect', 'gmrender-resurrect', 'gmrender-r
Warning: target file "./usr/lib/libbsa.so" is touched by more than one package: ['cypress_bsa'
Warning: target file "./usr/bin/app_av" is touched by more than one package: ['cypress_bsa', '
Warning: target file "./usr/bin/app_avk" is touched by more than one package: ['cypress_bsa',
Warning: target file "./usr/include/upnp/UpnpIntTypes.h" is touched by more than one package:
```

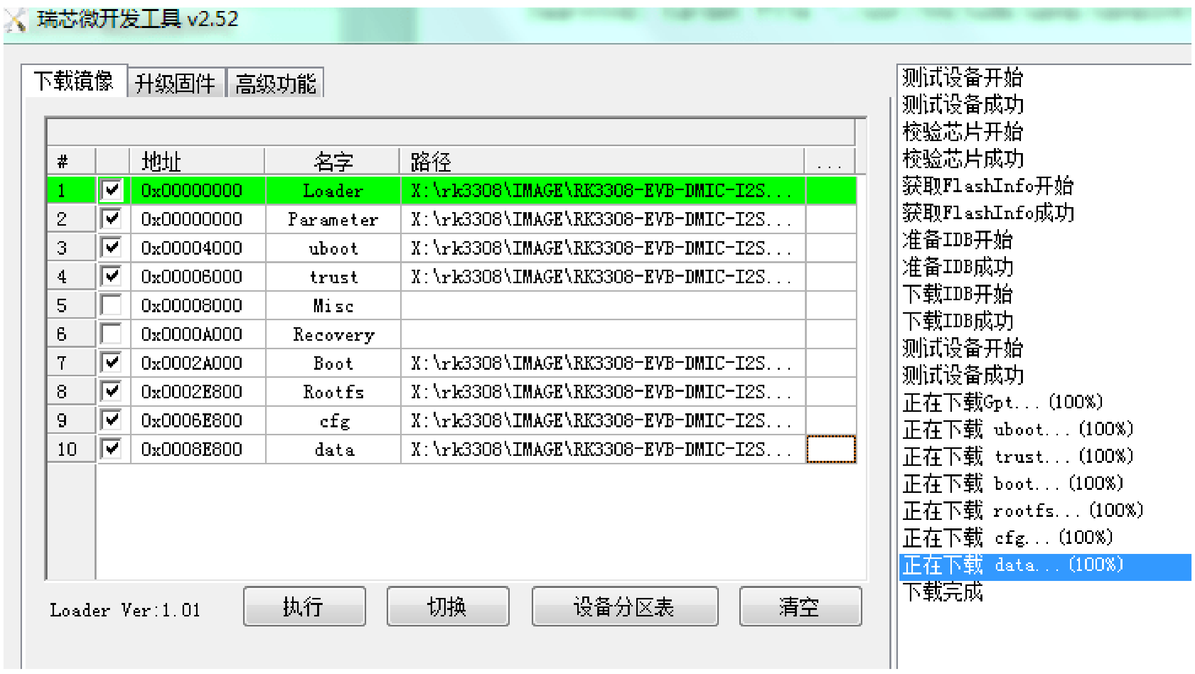
The generated rootfs is in buildroot\output\rockchip_rk3308_release\images\rootfs.squashfs, replace IMAGE\RK3308-EVB-DMIC-I2S-V10_20180515.1456_RELEASE_TEST\IMAGES\rootfs.img with this file.

The RK3308-EVB-DMIC-I2S-V10_20180515.1456_RELEASE_TEST folder is generated automatically, please refer to the new folder generated at that time.

After the compilation is complete, we can burn the code to run, please refer to chapter 4.2 Run.

4.2 Run

Use tools\windows\AndroidTool\AndroidTool_Release\AndroidTool.exe to burn the generated BIN into RK3308:



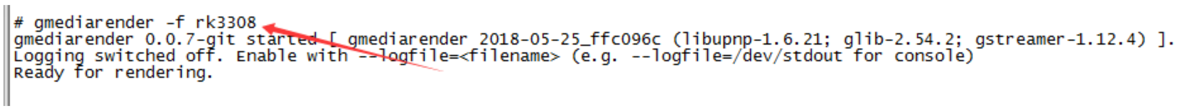
#		地址	名字	路径	...
1	✓	0x00000000	Loader	X:\rk3308\IMAGE\RK3308-EVB-DMIC-I2S...	
2	✓	0x00000000	Parameter	X:\rk3308\IMAGE\RK3308-EVB-DMIC-I2S...	
3	✓	0x00004000	uboot	X:\rk3308\IMAGE\RK3308-EVB-DMIC-I2S...	
4	✓	0x00006000	trust	X:\rk3308\IMAGE\RK3308-EVB-DMIC-I2S...	
5		0x00008000	Misc		
6		0x0000A000	Recovery		
7	✓	0x0002A000	Boot	X:\rk3308\IMAGE\RK3308-EVB-DMIC-I2S...	
8	✓	0x0002E800	Rootfs	X:\rk3308\IMAGE\RK3308-EVB-DMIC-I2S...	
9	✓	0x0006E800	cfg	X:\rk3308\IMAGE\RK3308-EVB-DMIC-I2S...	
10	✓	0x0008E800	data	X:\rk3308\IMAGE\RK3308-EVB-DMIC-I2S...	

```
测试设备开始
测试设备成功
校验芯片开始
校验芯片成功
获取FlashInfo开始
获取FlashInfo成功
准备IDB开始
准备IDB成功
下载IDB开始
下载IDB成功
测试设备开始
测试设备成功
正在下载Gpt... (100%)
正在下载 uboot... (100%)
正在下载 trust... (100%)
正在下载 boot... (100%)
正在下载 rootfs... (100%)
正在下载 cfg... (100%)
正在下载 data... (100%)
下载完成
```

Start RK3308 after burning is completed.

If the SDK starts up automatically to start DLNA, please refer to 4.3 to test the DLNA function.

If the SDK does not start DLNA automatically after booting up, you need to manually start DLNA as follows (The premise is connected to the network, otherwise DLNA will fail to start)

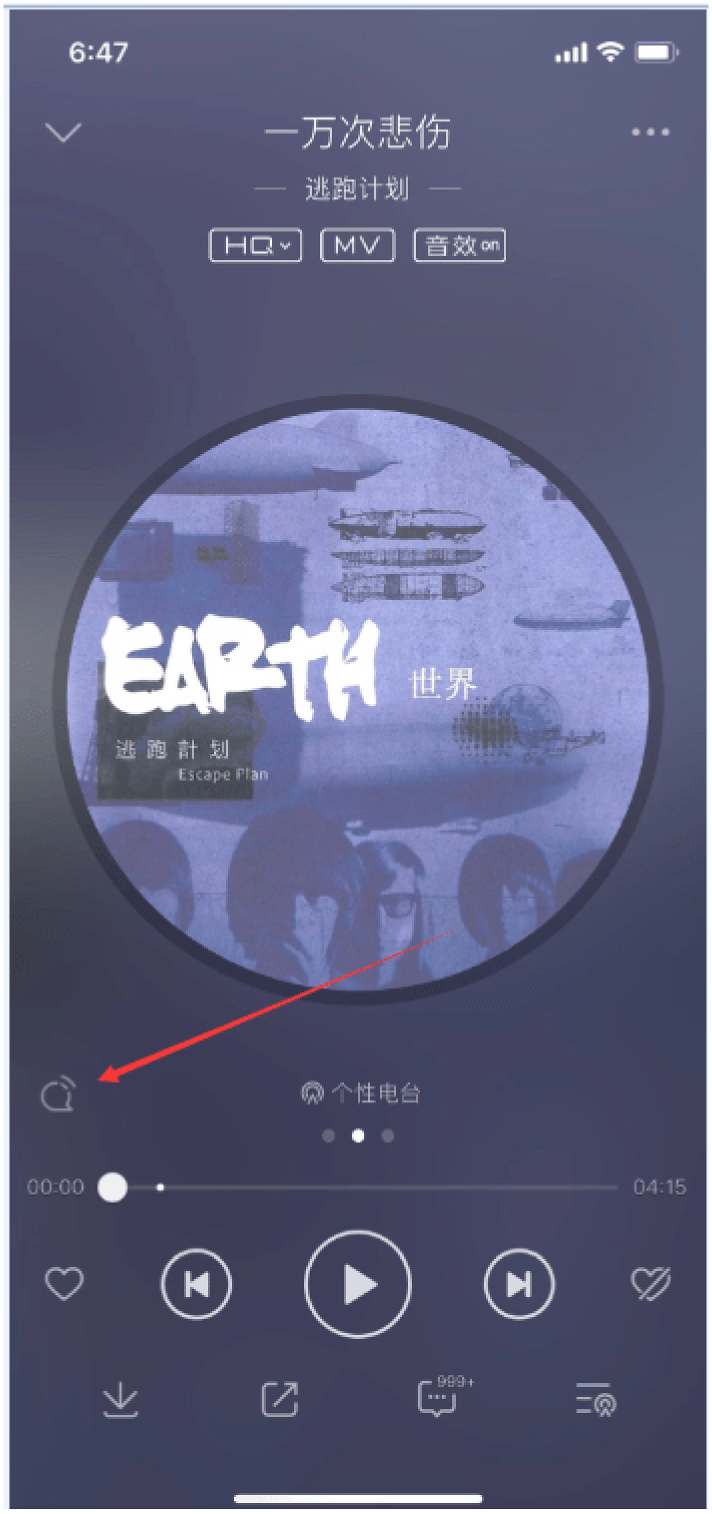


```
# gmediarender -f rk3308
gmediarender 0.0.7-git started [ gmediarender 2018-05-25_ffc096c (libupnp-1.6.21; glib-2.54.2; gstreamer-1.12.4) ].
Logging switched off. Enable with --logfile=<filename> (e.g. --logfile=/dev/stdout for console)
Ready for rendering.
```

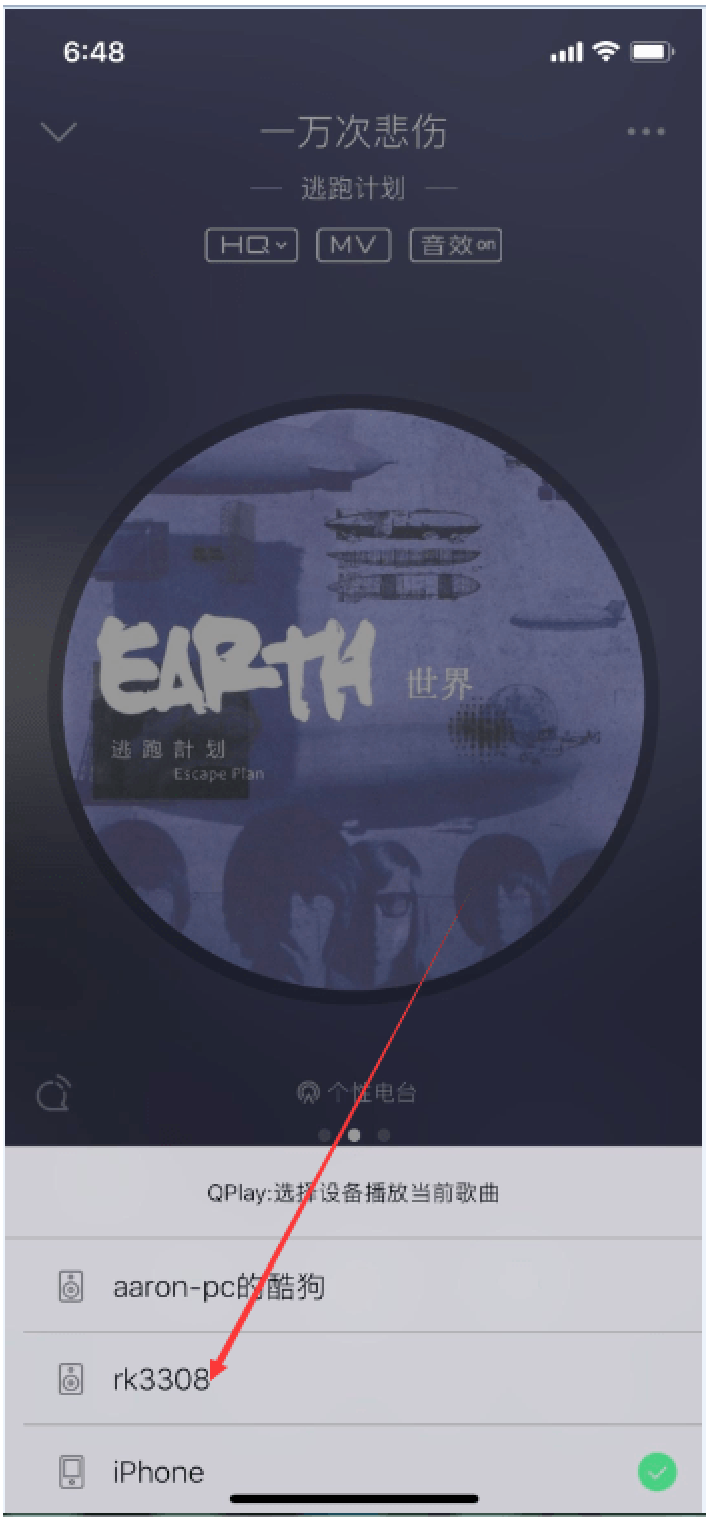
If Ready for rendering indicates appears, indicates the startup is successful and waiting for the phone to connect (rk3308 will be found by the phone as the name of DMR). For the next steps, please refer to 4.3

4.3 Test

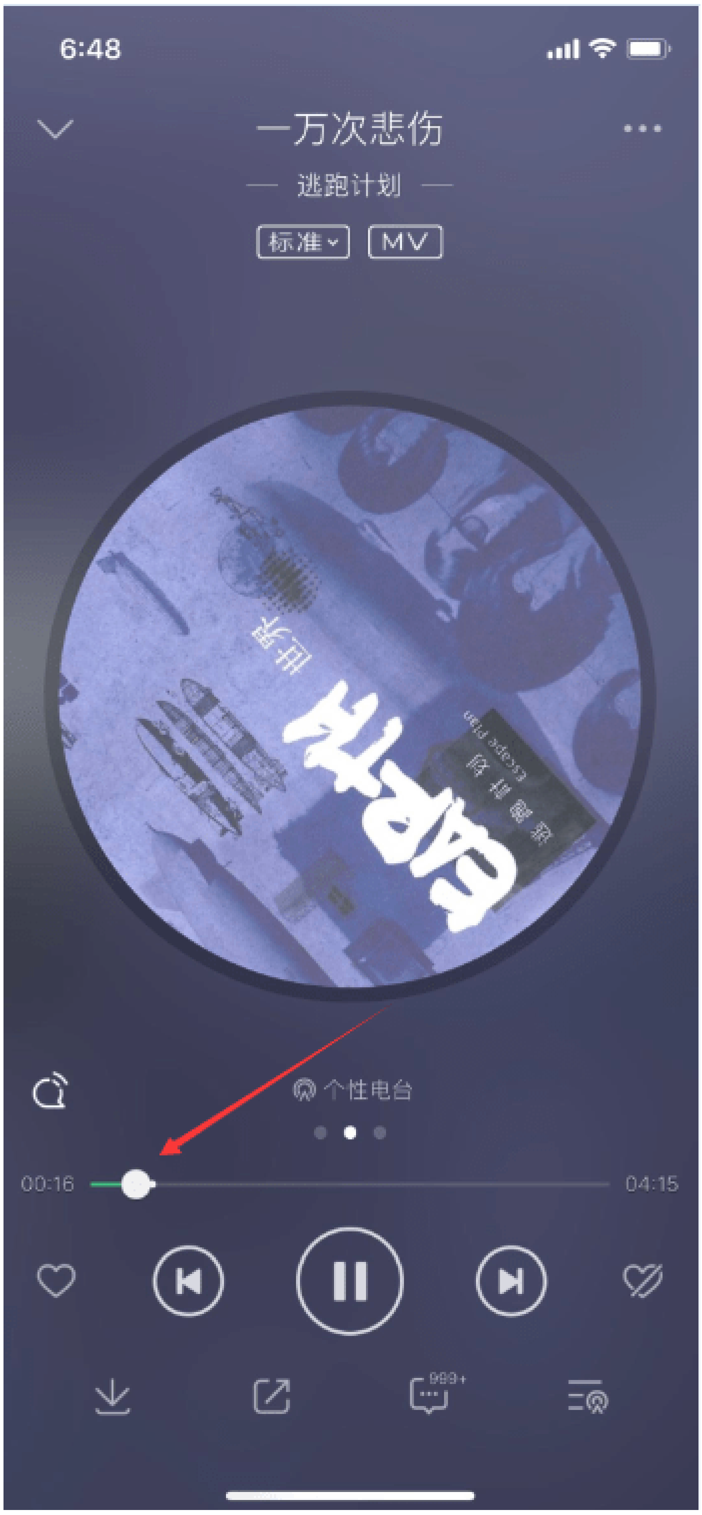
Connect a mobile phone to the local area network (it should be in the same network segment as RK3308), open the mobile QQ player, and select a song.



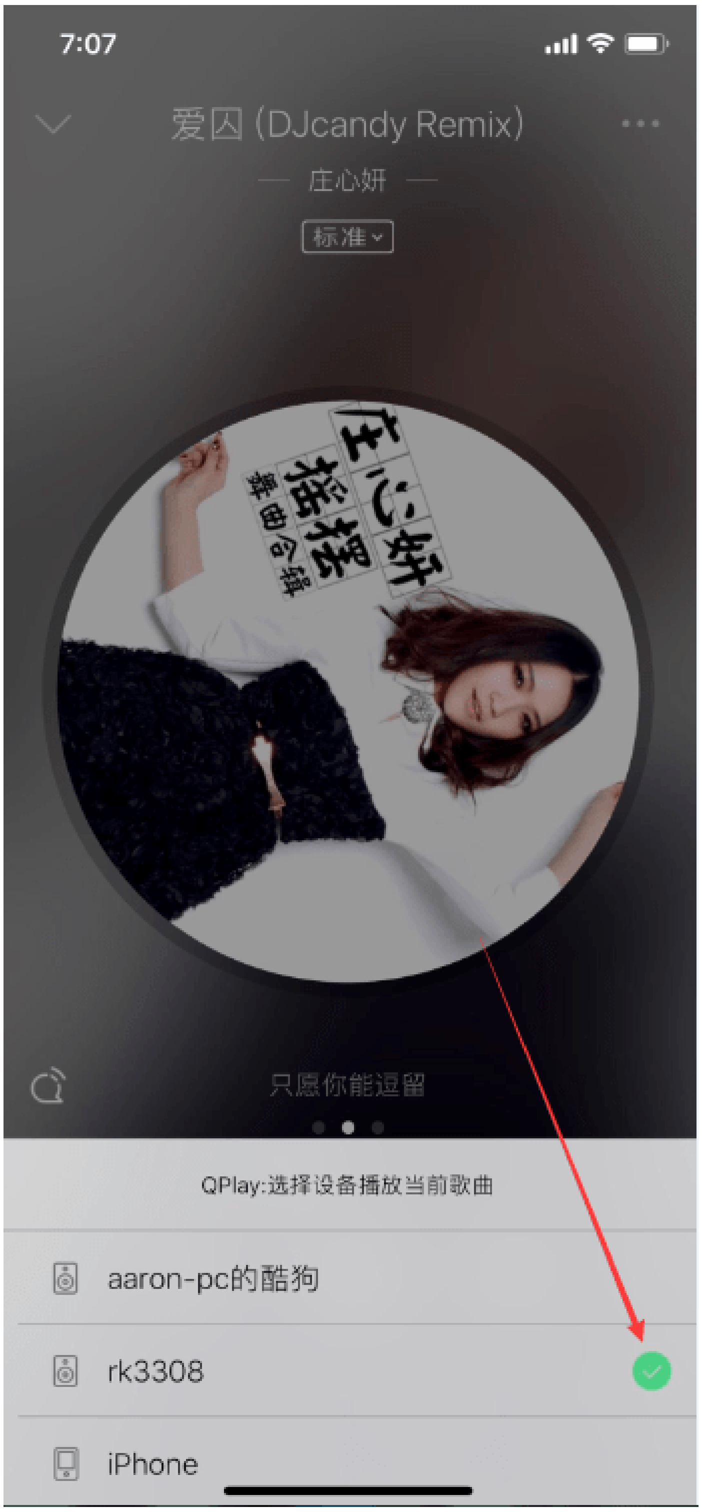
See the icon pointed by the red arrow in the above figure, click to open the following figure:



Rk3308 will appear. After selection, the song will be pushed to RK3308, and the phone will display progress bar:



During playback, you can adjust the volume, switch the upper and lower songs, and click again, you will see that the current playback device is RK3308, as shown in the figure below:


7:07

爱囚 (DJcandy Remix)

— 庄心妍 —

标准

只愿你能逗留

QPlay:选择设备播放当前歌曲

aaron-pc的酷狗

rk3308 ✓

iPhone