

DLNA开发指南

文件标识: RK-KF-YF-380

发布版本: V1.0.2

日期: 2020-08-11

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司(“本公司”, 下同)不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自拥有者所有。

版权所有 © 2020 瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: www.rock-chips.com

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: fae@rock-chips.com

前言

概述

本文主要介绍基于Buildroot DLNA开发指南。

产品版本

芯片名称	内核版本
RK3308	4.4

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V1.0.0	SCH	2018-05-28	初始版本
V1.0.1	Ruby Zhang	2019-05-29	格式修正
V1.0.2	Ruby Zhang	2020-08-11	更新公司名称转换成md格式

目录

DLNA开发指南

1. DLNA概述
2. DLNA重要概念
3. DLNA框架
4. 开发指引
 - 4.1 编译
 - 4.1.1 版本确认
 - 4.1.2 配置
 - 4.1.3 编译
 - 4.2 运行
 - 4.3 测试

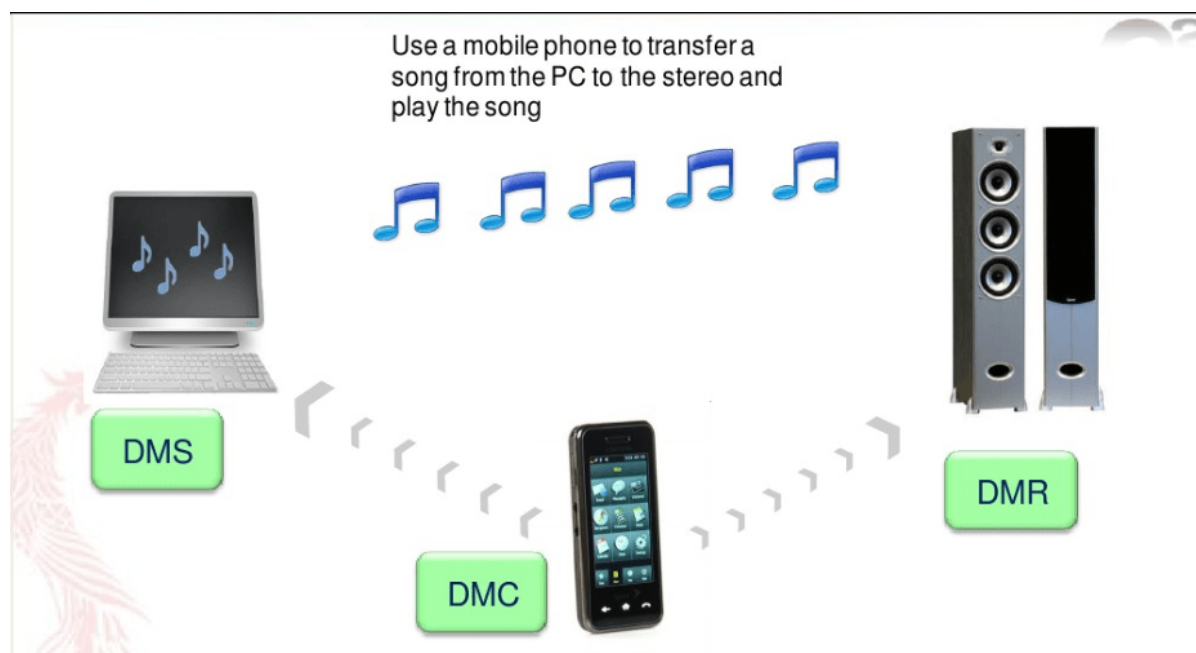
1. DLNA概述

DLNA的全称是DIGITAL LIVING NETWORK ALLIANCE(数字生活网络联盟)。

DLNA 成立于2003 年6 月24 日, 其前身是DHWG (Digital Home Working Group 数字家庭工作组), 由 Sony、Intel、Microsoft等发起成立、旨在解决个人PC , 消费电器, 移动设备在内的无线网络和有线网络的互联互通, 使得数字媒体和内容服务的无限制的共享和增长成为可能。DLNA的口号是Enjoy your music, photos and videos, anywhere anytime。

DLNA 宣布组织在2017年1月15日正式解散, 未来不会再更新DLNA标准。

DLNA将其整个应用规定成5个功能组件。从下到上依次为: 网络互连, 网络协议, 媒体传输, 设备的发现控制和管理, 媒体格式, 以下是DLNA的一个应用场景(实际场景远不止这些)



2. DLNA重要概念

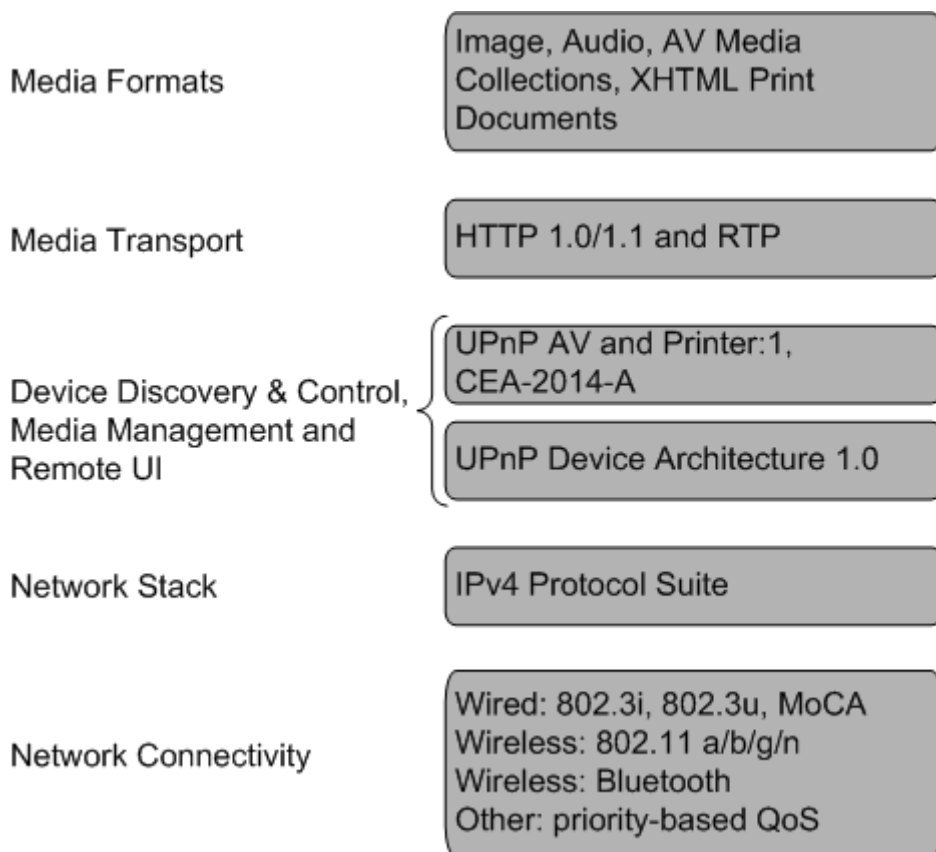
- Home NetWork Device(HND)。这类设备指家庭设备, 具有比较大的尺寸及较全面的功能, 主要与移动设备区别开来, 下属5类设备:
 1. Digital Media Server(DMS)。数字媒体服务器, 提供媒体获取、记录、存储和输出功能。同时, 内容保护功能是对DMS的强制要求。DMS总是包含DMP的功能, 并且肯能包含其他智能功能, 包括设备/用户服务的管理; 丰富的用户界面; 媒体管理/收集和分发功能。DMS的例子有PC、数字机顶盒(附带联网, 存储功能)和摄像机等等。
 2. DMP。数字媒体播放器。能从DMS/M-DMS上查找并获取媒体内容并播放和渲染显示。比如智能电视、家庭影院等。
 3. DMC。数字媒体控制器, 查找DMS的内容并建立DMS与DMR之间的连接并控制媒体的播放。如遥控器。
 4. DMR。数字媒体渲染设备。通过其他设备配置后, 可以播放从DMS上的内容。与DMP的区别在于DMR只有接受媒体和播放功能, 而没查找有浏览媒体的功能。比如显示器、音箱等。
 5. DMP。数字媒体打印机, 提供打印服务。网络打印机, 一体化打印机就属于DMP。

- Mobile Handheld Devices(MHD)手持设备。相比家庭设备，手持设备的功能相对简化一些，支持的媒体格式也会不同。
 1. M-DMS。与DMS类似，如移动电话，随身音乐播放器等。
 2. M-DMP。与DMP类似。比如智能移动电视。
 3. M-DMD。移动多媒体下载设备。如随身音乐播放器，车载音乐播放器和智能电子相框等
 4. M-DMU。移动多媒体下载设备。如摄像设备和手机等。
 5. M-DMC。与DMC类似。P如DA，智能遥控器。手持设备没有定义M-DMR，因为手持设备会讲究便利性，会附加查找控制功能，要不然就只是普通的移动电视或收音机了。
- Networked Infrastructure Devices (NID) 联网支持设备。
 1. Mobile Network Connectivity Function (M-NCF)。移动网络连接功能设备。提供各种设备接入移动网络的物理介质。DLNA的希望是全部实现无线化。
 2. Interoperability Unit (MIU)媒体交互设备。提供媒体格式的转换以支持各种设备需要。

3. DLNA框架

DLNA架构是个互联系统，因此在逻辑上它也类似OSI（Open System Interconnection，开放系统互连）七层网络模型。

DLNA架构分为如下图7个层次：



1. NetWorking Connectivity 网络互联方式:包括物理连接的标准，有有线的，比如符合IEEE802.3标准的Ethernet；有无线的，比如符合IEEE802.11a/g标准的WiFi，能做到54Mbps，蓝牙(802.15)等技术都很成熟。现在OFDM和MIMO(802.11n)已经能做到300Mbps了，早就超过比较普及的100Mbps的Ethernet了，只不过产品还没有普及，以后肯定会用到。
2. NetWorking Stack 网络协议栈：DLNA的互联传输基本上是在IPV4协议簇的基础上的。用TCP或者UDP来传都可以。这一层相当于OSI网络层。

3. Device Discovery&Control 设备发现和控制。这个层次是比较essential的，是DLNA的基础协议框架。DLNA用UPnP协议来实现设备的发现和控制。这一部分可以看一下<http://upnp.org/sdcp-s-and-certification/standards/device-architecture-documents/>里的文档。
4. Media Management媒体管理。媒体管理包括媒体的识别、管理、分发和记录（保存），UPnP AV Architecture:1 and UPnP Printer Architecture:1这两个属于UPnP的文档会说明如何进行媒体管理。
5. Media Transport 媒体传输：这一层用HTTP(HyperText Transfer Protocol)超文本传输协议。就是平时我们上网用的媒体传输协议。HTTP用TCP可靠传输，也有混合UDP方式的HTTP。现在HTTP的最新版本是HTTP1.1。可选协议是RTP。
6. Media Formats媒体格式。格式Formats在这里等同于编码格式Codec，平时我们说的编码格式比如Mpeg-2, AVC, x264就是视频编码格式；PCM, mp3(MPEG-2 Layer 3), aac, flac就是音频编码格式。而avi, rmvb, mkv这些是媒体封装格式，包含视频音频可能还有字幕流。比如一个常见的后缀为mkv的文件，它的视频Codec是x264，音频是aac，它的视音频编码属于Mpeg-4 Codec Family。

4. 开发指引

前3章我们对DLNA有个初步的了解，接下来我们来搭建一个DLNA环境实现 M-DMS（手机QQ音乐播放器）推送音乐到DMR(RK3308智能音箱)。

我们选择gmrender-resurrect开源代码来搭建DMR角色，找一个手机安装QQ播放器充当M-DMS角色。

4.1 编译

4.1.1 版本确认

编译之前我们先确认一下gmrender-resurrect和关联库libupnp的版本，确保版本符合一下要求：

gmrender-resurrect版本：33600ab663f181c4f4f5c48aba25bf961760a300

Libupnp版本：1.6.21

Buildroot的package的配置信息在Buildroot/package文件下相应的文件夹里，每个文件夹包含3个文件，分别是*.in, *.hash, *.mk, 以下是gmrender-resurrect 和 libupnp的配置文件截图：

```
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$
```

其中

*.in文件记录了makeconfig中的开关宏

*.hash文件记录了代码压缩包的HASH值和版本信息

*.mk文件记录了代码编译信息和版本信息

我们使用vi命令打开对应的文件，来确认版本信息，如下图：

```

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 配置

编译之前按照以下步骤进行配置

1. 设置环境变量，使用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. 配置menuconfig

```

sch@SYS3:~/rk3308$ make menuconfig
umask 0022 && make -C /home/sch/rk3308/buildroot o=/home/sch/rk3308/buildroot/output/rockchip_rk3308_release/Makefile
GEN      /home/sch/rk3308/buildroot/output/rockchip_rk3308_release/Makefile
/home/sch/rk3308/buildroot/output/rockchip_rk3308_release/.config - Buildroot 2018.02-rc3-00079-g...

```

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 --->

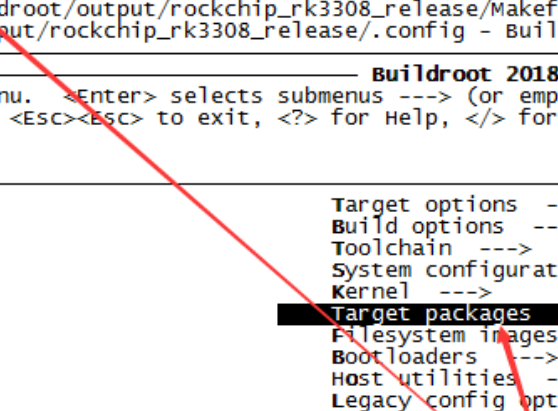
Target packages --->

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 --->

```

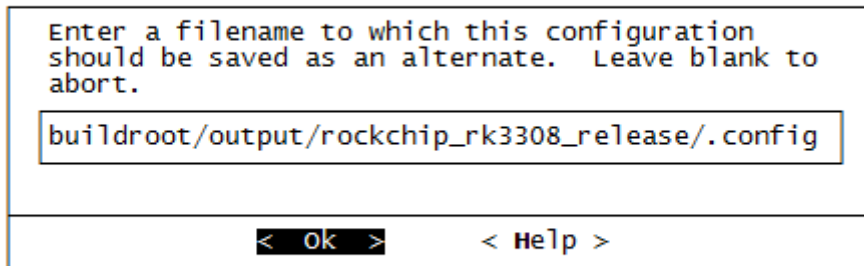
```

[] 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>** <Load>

如果gmrender-resurrect 未被选中（图为选中），按Y选中后save保存：



保存config之后还不能编译，需要使用make savedefconfig导出配置

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

成功导出配置后我们可以相关的编译，请参考1.4.1.3

4.1.3 编译

1. 如果没有编译过，请使用build.sh进行全编译

```
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
```

全编译之后，会在IMAGE目录下生成一个子目录IMAGE\RK3308-EVB-DMIC-I2S-

V10_20180515.1456_RELEASE_TEST\IMAGES，该目录下包含了RK3308所需要的全部IMG，如下图：

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

如果系统有全编译过，我们可以使用make gmrender-resurrect-rebuild只编译gmrender-resurrect及其相关库。

```
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_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/buildroot/output/rockchip_rk3308_release/host/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
```

gmrender-resurrect单独编译完成之后， 需要使用make命令来生成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_rk3308_release/build/gmrender-resurrect-33600ab6
>>> 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_release/build/gmrender-resurrect-33600ab6
warning: target file "/usr/include/upnp/ixmldebug.h" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/include/upnp/upnpString.h" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/share/info/dir" is touched by more than one package: ['readline', 'libupnp', 'gmrender-resurrect']
warning: target file "/usr/bin/app_hs" is touched by more than one package: ['cypress_bsa', 'libupnp', 'gmrender-resurrect']
warning: target file "/usr/include/upnp/upnpdebug.h" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/sbin/i2cset" is touched by more than one package: ['busybox', 'i2c-tools', 'gmrender-resurrect']
warning: target file "/usr/include/upnp/upnpGlobal.h" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/include/upnp/upnpTools.h" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/sbin/i2cget" is touched by more than one package: ['busybox', 'i2c-tools', 'gmrender-resurrect']
warning: target file "/usr/share/man" is touched by more than one package: ['readline', 'iw', 'gmrender-resurrect']
warning: target file "/usr/lib/libthreadutil.so" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/include/upnp/ixml.h" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/lib/libupnp.so" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/include/upnp/TimerThread.h" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/lib/libixml.so.2" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/lib/libthreadutil.la" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/include/upnp/upnpInet.h" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/include/upnp/ithread.h" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
warning: target file "/usr/bin/gmediarender" is touched by more than one package: ['gmrender-resurrect', 'gmrender-resurrect', 'gmrender-resurrect', 'gmrender-resurrect']
warning: target file "/usr/lib/libbsa.so" is touched by more than one package: ['cypress_bsa', 'gmrender-resurrect']
warning: target file "/usr/bin/app_av" is touched by more than one package: ['cypress_bsa', 'gmrender-resurrect']
warning: target file "/usr/bin/app_avk" is touched by more than one package: ['cypress_bsa', 'gmrender-resurrect']
warning: target file "/usr/include/upnp/upnpIntTypes.h" is touched by more than one package: ['libupnp', 'gmrender-resurrect']
```

生成的rootfs在buildroot\output\rockchip_rk3308_release\images\rootfs.squashfs, 用该文件替换掉IMAGE\RK3308-EVB-DMIC-I2S-V10_20180515.1456_RELEASE_TEST\IMAGES\rootfs.img。

RK3308-EVB-DMIC-I2S-V10_20180515.1456_RELEASE_TEST该文件夹是自动生成的，请以当时生成的文件夹为准。

编译完成之后，我们就可以烧写代码进行运行了，请参考1.4.2运行。

4.2 运行

使用工具tools\windows\AndroidTool\AndroidTool_Release\AndroidTool.exe将生成的BIN烧写到RK3308当中：



烧写完成后启动RK3308。

如果SDK开机自动启动DLNA，请参考1.4.3测试DLNA的功能。

如果SDK开机没有自动启动DLNA，需要按如下方式手动启动DLNA(前提是已经连接网络，否则DLNA会启动失败)

```
# 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.
```

看到Ready for rendering说明启动成功等待手机连接（rk3308将作为DMR的名称被手机看到），接下来的步骤请参考1.4.3

4.3 测试

将手机接入局域网（需要和RK3308同一个网段），打开手机QQ播放器，选择一首歌曲。




看到上图红色箭头指定的图标，点击打开如下图：



将会出现rk3308, 选择后这首歌曲将被推送到RK3308上，手机进入显示播放的进度条：



播放的过程中，你可以调节音量，切换上下曲，再次单击 可以看到当前播放设备是RK3308,如下图：

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