Rockchip Debian Developer Guide

ID: RK-KF-YF-913

Release Version: V1.0.1

Release Date: 2022-03-10

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 introduces how to use the official released Debian system to build and adapt related hardware functions based on Rockchip arm platforms.

Supported Chipset

Chipset	Supproted Debian	Kernel Version
RK3588	11	5.10
RK3568	10	4.19
RK3566	10	4.19
RK3399	10	4.4、4.19
RK3399PRO	10	4.4、4.19
RK3326	10	4.4、 4.19
PX30	10	4.4、 4.19
RK3288	10	4.4、4.19
RK3328	10	4.4
RK312X	10	4.4

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Date	Version	Author	Change Description
2021-12-30	V1.0.0	Caesar Wang	Initial version
2022-03-10	V1.0.1	Ruby Zhang	Update the format of the document

Contents

1. Debian Overview

1.1 What is Debian

Debian is a completely free and open Linux operating system that is widely used on various devices. The reasons for choosing Debian are as follows:

- Debian is free software

  Debian is made up of free and open source code and will always remain 100% free. Everyone is free to use, modify, and release. You can perform secondary development based on the Debian system built by Rockchip.

- Debian is a stable and secure operating system based on Linux.

  Debian is an operating system that is widely used on a variety of devices, including laptops, desktops, and servers. Its stability and reliability have been loved by users since 1993. We provide sensible default configurations for each package. Debian developers try to provide security updates for all packages during their lifetime as far as possible.

- Debian has widely hardware support.

  Most hardware is already supported by Linux kernel. Dedicated hardware drivers can also be used when free software does not provide sufficient support. At present, Rockchip RK3588/RK3568/RK3566/RK3399/RK3288 and other chips have been adapted and supported.

- Debian provides smooth updates.

  Debian is known for easy and smooth updates during its release cycle, not only that, but also it can easily upgrade to the next major release version. At present, Rockchip has been upgraded from Debian Stretch (9) to Debian Buster (10) and Bullseye (11) versions.

- Debian is the seed and base for many other release versions.

  Many popular Linux released versions, such as Ubuntu, Knoppix, PureOS, SteamOS, and Tails, have chosen Debian as their software base. Debian provides all the tools, therefore, everyone can use the software package that can meet their own needs to expand the software package in Debian libraries.

- Debian project is a community.

  Debian is not just a Linux operating system. The software is co-worked by hundreds of volunteers from all over the world. You can become a part of the Debian community even if you are not a programmer or system administrator.

1.2 Supported Debian System Versions

Versions	Supported Architectures	schedule	Current status
Debian 9 “Stretch”	armhf and arm64	July 6, 2020 to June 30, 2022	No longer maintain
Debian 10 “Buster”	armhf and arm64	July, 2022 to June, 2024	Under maintenance
Debian 11 “Bullseye”	armhf and arm64	July, 2024 to June, 2026	Under development

More Debian Long Term Support Version time, please refer to the official website.

2. Debian Quick Strat Guide

2.1 Environment Setup

It is recommended to use Ubuntu 21.04 for compilation. Other Linux versions may need to adjust the software package accordingly. In addition to the system requirements, there are other hardware and software requirements. Hardware requirements: 64-bit system, hard disk space should be greater than 40G. If you do multiple builds, you will need more hard drive space

Software requirements: Ubuntu 21.04 system:

Please install software packages with below commands to setup SDK building environment:

```
sudo apt-get install repo git ssh make gcc libssl-dev liblz4-tool \
expect g++ patchelf chrpath gawk texinfo chrpath diffstat binfmt-support \
qemu-user-static live-build bison flex fakeroot cmake gcc-multilib g++-multilib
unzip \
device-tree-compiler ncurses-dev \
```

It is recommended to use the Ubuntu21.04 system or higher version for development. If an error is reported during compilation, you can install the corresponding software package according to the error message.

2.2 Get the Source Code

The source code can be get from the SDK released by Rockchip code server, which is located in the project `<SDK>/Debian` directory.

2.3 Build

Enter the SDK project and build directly

```
./build.sh debian
```

Or go to the debian/ directory:

```
cd debian/
```

Please refer to the current directory readme.md for later compilation and Debian firmware generation.

(1) Building base Debian system

```
sudo apt-get install binfmt-support qemu-user-static live-build
sudo dpkg -i ubuntu-build-service/packages/*
sudo apt-get install -f
```

3. Debian Directory Structure

```
debian
├── mk-base-debian.sh ##Get Debian base package and build
├── mk-image.sh ##Packag and generate ext4 firmware
├── mk-rootfs-buster/bullseye.sh ##Adapt to Rockchip related hardware
acceleration package
├── mk-rootfs.sh ##Points to the specific Rootfs version, there are currently two
versions of Buster and Bullseye.
├── overlay ##Adapt to the commonly used configuration file of Rockchip platform
├── overlay-debug ##Debugging tools commonly used by the system
├── overlay-firmware ##Storage of some device firmware, such as wifibt/dp, etc.
├── packages ## Contains pre-built packages for armhf arm64 system adaptation for
hard acceleration
├── packages-patches ##Pre-built packages, based on official patches
├── readme.md ## Documentation guide
└── ubuntu-build-service ##Get the Debian released version,dependent packages and
custom installation related packages from official
```

The content of the entire directory structure is implemented through shell scripts to obtain the source code of the Linux Debian released versions, build and install the operating system adapted to Rockchip hard acceleration package.

4. Debian Live-build Usage Guide

Live build is a group of scripts used to build live system images. The idea behind live build is a tool kit that uses a configuration directory to fully automate and customize all aspects of building live images.

4.1 Related Commands

- lb config

Create auto and config directories and related configuration files in the current directory, and run the auto/config script.

- lb clean

run auto/clean scripts

- lb build

Build system images according to various configuration scripts in the config directory

4.2 Software Source Settings

- The first way:

```
$ lb config --mirror-bootstrap http://mirrors.ustc.edu.cn/debian \
--mirror-chroot-security http://mirrors.ustc.edu.cn/debian-security/ \
--mirror-chroot-backports http://mirrors.ustc.edu.cn/debian-backports/
```

chroot mirror: --mirror-chroot, use the value of --mirror-bootstrap by default or create a config/archives/your-repository.list.chroot file with the content of source address. Sources will be added to the /etc/apt/sources.list.d/ directory of the live system.

- The Second way:

```
$ lb config --mirror-binary http://mirrors.ustc.edu.cn/debian \
--mirror-binary-security http://mirrors.ustc.edu.cn/debian-security/
```

Or create a config/archives/your-repository.list.binary file with the content of source address.

4.3 Customize System Packages

- The first way:

Place the required package list in the customization/package-lists directory and name it XXX.list.chroot or XXX.list.binary.

- The Second way:

With --package-lists "XXX", the specified package list under /usr/share/live/build/package-lists/ will be used.

After executing lb config, four configuration files, binary, bootstrap, chroot, and common in the config directory will be generated according to the parameters in this script. lb build reads these four configuration files, so you can also make modifications to the parameters in these four files after lb config.

Configuration parameters is in auto/config, for example:

```
set -e

echo "I: create configuration"
export LB_BOOTSTRAP_INCLUDE="apt-transport-https gnupg"
lb config \
 --mirror-bootstrap "http://mirrors.ustc.edu.cn/debian" \
 --mirror-chroot "http://mirrors.ustc.edu.cn/debian" \
 --mirror-chroot-security "http://mirrors.ustc.edu.cn/debian-security" \
 --mirror-binary "http://mirrors.ustc.edu.cn/debian" \
 --mirror-binary-security "http://mirrors.ustc.edu.cn/debian-security" \
```

```
 --apt-indices false \
 --apt-recommends false \
 --apt-secure false \
 --architectures arm64 \
 --archive-areas 'main contrib non-free' \
 --backports false \
 --binary-filesystem ext4 \
 --binary-images tar \
 --bootappend-live "hostname=linaro-alip username=linaro" \
 --bootloader "syslinux" \
 --bootstrap-qemu-arch arm64 \
 --bootstrap-qemu-static /usr/bin/qemu-aarch64-static \
 --cache false \
 --chroot-filesystem none \
 --compression gzip \
 --debootstrap-options "--variant=minbase --include=apt-transport-https,gnupg" \
 --distribution bullseye \
 --gzip-options '-9 --rsyncable' \
 --iso-publisher 'Linaro; http://www.linaro.org/; linaro-dev@lists.linaro.org' \
 --iso-volume 'Linaro Bullseye $(date +%Y%m%d-%H:%M)' \
 --linux-flavours none \
 --linux-packages none \
 --mode debian \
 --security true \
 --system normal \
 --updates true
```

4.3.1 Customized Directory

The customized directory and the files in it can be placed in the corresponding include directory of config/

config/binary_local-includes (take the root directory of the generated image as the root directory)
config/chroot_local-includes (take the root directory of the target system as the root directory)

4.3.2 HOOKS

The scripts in config/hooks are run after each stage of live-build has completed.
config/binary_local-hooks
config/chroot_local-hooks

A new version of live-build will get patches from the live/normal directories.

```
customization/hooks/live/
├── 0001-setup_user_linaro.binary
├── 0002-add_linaro_to_groups.binary
├── 0003-check_sudoers_for_admin.binary
├── 0021-silence-systemd.binary
├── 0022-disable-systemd-services.binary
├── 0023-lightdm-autologin.binary
└── 0098-resolvconf.binary
```

5. The Pre-built Packages Integrated in Debian Introduction

```
packages
├── armhf/arm64
│   ├── mpp
│   ├── libv4l
│   ├── chromium
│   ├── glmark2
│   ├── gst-plugins-bad1.0
│   ├── gst-plugins-base1.0
│   ├── gst-plugins-good1.0
│   ├── gst-rkmpp
│   ├── libdrm
│   ├── libdrm-cursor
│   ├── libmali
│   ├── openbox
│   ├── pcmanfm
│   ├── rga
│   ├── rkaiq
│   ├── rkisp
│   ├── rkwifibt
│   └── xserver
```

5.1 mpp

The Media Process Platform (MPP) provided by Rockchip is a general media processing software platform for Rockchip chipset. The platform shields the application software from the complex underlying processing related to chips, and its purpose is to shield the difference between different chips and provide users with a unified video media processing interface (The abbreviation of Media Process Interface is MPI).

Features provided by MPP include:

- Video decoding

  H.265/H.264/H.263/VP9/VP8/MPEG-4/MPEG-2/MPEG-1/VC1/MJPEG

- Video encoding

  H.264/VP8/MJPEG

- Video processing

  Video copying, scaling, color space conversion, field video deinterlace

Please refer to the MPP development documentation for more information.

The pre-built packages integrated in Debian are as follows:

```
mpp/
├── librockchip-mpp-dev_1.5.0-1_arm64.deb
├── librockchip-mpp1-dbgsym_1.5.0-1_arm64.deb
├── librockchip-mpp1_1.5.0-1_arm64.deb
├── librockchip-vpu0-dbgsym_1.5.0-1_arm64.deb
├── librockchip-vpu0_1.5.0-1_arm64.deb
├── rockchip-mpp-demos-dbgsym_1.5.0-1_arm64.deb
└── rockchip-mpp-demos_1.5.0-1_arm64.deb
```

5.2 libv4l

The v4l2 plugin that connects Chromium browser and mpp to implement hard decode.
The pre-built packages:

```
/libv4l/
...
├── libv4l-dev_1.16.3-3_arm64.deb
├── libv4l-rkmpp-dbgsym_1.3.2-1_arm64.deb
├── libv4l-rkmpp_1.3.2-1_arm64.deb
├── libv4l2rds0-dbgsym_1.16.3-3_arm64.deb
├── libv4l2rds0_1.16.3-3_arm64.deb
├── libv4lconvert0-dbgsym_1.16.3-3_arm64.deb
├── libv4lconvert0_1.16.3-3_arm64.deb
├── qv4l2-dbgsym_1.16.3-3_arm64.deb
├── qv4l2_1.16.3-3_arm64.deb
├── v4l-utils-dbgsym_1.16.3-3_arm64.deb
└── v4l-utils_1.16.3-3_arm64.deb
```

5.3 Chromium

```
chromium/
├── chromium-x11-dbgsym_91.0.4472.164_arm64.deb
└── chromium-x11_91.0.4472.164_arm64.deb
```

Chromium browser supports H264\VP8\VP9 and other video formats, but does not support H265. At present, Debian has integrated the support of Chromium video hardware decoding, using customized chromium+v4l2 plugin+mpp for efficient hardware decoding.

The main modifications of customized Chromium are as follows:

- Modify chromium to enable v4l2 vda support, and related patches
- add v4l2 mpp plugin

Disadvantages:
a. Only support vp8, h264, vp9
b. You need to modify and compile chromium (after tuning the complete compilation process of yocto)
Currently, we provide chromium wayland patch, support for yocto, buildroot, and a few customers are using it

- Chromium version

root@linaro-alip:~# chromium --version
Chromium 91.0.4472.164 stable

- Command to test hardware decoding:

Use the following command to test:

```
chromium --no-sandbox file:///usr/local/test.mp4
```

The test script is in /usr/local/bin/test_dec-chromium.sh:

```
root@linaro-alip:~# test_dec-chromium.sh
[2588:2588:0214/104846.535688:ERROR:gpu_init.cc(440)] Passthrough is not
supported, GL is egl
...
```

Linux4.19/5.10 can use the following command to check whether to call the hardware decoding:

```
echo 0x100 > /sys/module/rk_vcodec/parameters/mpp_dev_debug
```

Linux4.4 can use the following command to check whether to call the hardware decoding:

```
echo 0x4 > /sys/module/rk_vcodec/parameters/debug
```

- How to debug

If you encounter some problems, turn on the following switches to get more logs for debugging.

export mpi_debug=1
export mpp_debug=1
export h264d_debug=1

- Others

For more chromium information, you can enter chrome://about in the URL to get.

```
List of Chrome URLs
chrome://about
...
chrome://flags
chrome://gcm-internals
chrome://gpu
chrome://help
chrome://histograms
chrome://history
chrome://indexeddb-internals
chrome://inspect
chrome://interstitials
chrome://invalidations
chrome://settings
chrome://version
chrome://webrtc-internals
chrome://webrtc-logs
List of chrome://internals pages
chrome://internals/web-app

For Debug
The following pages are for debugging purposes only. Because they crash or hang
the renderer, they're not linked directly; you can type them into the address bar
if you need them.

chrome://badcastcrash/
chrome://memory-exhaust/
chrome://memory-pressure-critical/
chrome://memory-pressure-moderate/
...
chrome://quit/
```

```
chrome://restart/
```

5.4 glmark2

```
glmark2/
├── glmark2-data_2021.02+ds-1_all.deb
├── glmark2-es2-x11-dbgsym_2021.02+ds-1_arm64.deb
└── glmark2-es2-x11_2021.02+ds-1_arm64.deb
```

Glmark2 is an open source benchmarking program for OpenGL 2.0 and ES 2.0, generally used to benchmark GPU.

Open source code of Glmark2 detailed test have been integrated into Debian

```
usr/local/bin/test_glmark2_*.sh
├── test_glmark2_fullscreen.sh ### full screen test
├── test_glmark2_normal.sh ### Default 800x600 resolution test
├── test_glmark2_offscreen.sh ### Offscreen test

root@linaro-alip:~# test_glmark2_normal.sh
/usr/local/bin/test_glmark2_normal.sh: line 36: warning: command substitution:
ignored null byte in input
performance
arm_release_ver of this libmali is 'g2p0-01eac0', rk_so_ver is '4'.
=======================================================
    glmark2 2021.02
=======================================================
    OpenGL Information
    GL_VENDOR:     ARM
    GL_RENDERER:   Mali-G52
    GL_VERSION:    OpenGL ES 3.2 v1.g2p0-01eac0.327c41db9c110a33ae6f67b4cc0581c7
=======================================================
[build] use-vbo=false:
---
```

For more usage, please use help command:

```
root@linaro-alip:~# glmark2-es2 --help
A benchmark for Open GL (ES) 2.0

Options:
  -b, --benchmark BENCH  A benchmark or options to run: '(scene)?(:opt1=val1)*'
                         (the option can be used multiple times)
  -f, --benchmark-file F Load benchmarks to run from a file containing a
                         list of benchmark descriptions (one per line)
                         (the option can be used multiple times)
      --validate         Run a quick output validation test instead of
                         running the benchmarks
      --data-path PATH   Path to glmark2 models, shaders and textures
                         Default: /usr/share/glmark2
      --frame-end METHOD How to end a frame [default,none,swap,finish,readpixels]
      --off-screen       Render to an off-screen surface
      --visual-config C  The visual configuration to use for the rendering
                         target: 'red=R:green=G:blue=B:alpha=A:buffer=BUF'.
```

```
                        The parameters may be defined in any order, and any
                        omitted parameters assume a default value of '1'
    --reuse-context     Use a single context for all scenes
                        (by default, each scene gets its own context)
-s, --size WxH          Size of the output window (default: 800x600)
    --fullscreen        Run in fullscreen mode (equivalent to --size -1x-1)
-l, --list-scenes       Display information about the available scenes
                        and their options
    --show-all-options Show all scene option values used for benchmarks
                        (only explicitly set options are shown by default)
    --run-forever       Run indefinitely, looping from the last benchmark
                        back to the first
    --annotate          Annotate the benchmarks with on-screen information
                        (same as -b :show-fps=true:title=#info#)
-d, --debug             Display debug messages
-h, --help              Display help
```

5.5 gst-plugins-base1.0

Add support for dma buffer and rga/gpu graphics acceleration adaptation, based on gstreamer's official gst-plugins-base 1.14.4 and 1.18.5 versions.

gst-plugins-base1.0-1.14.4 and 1.18.5 patches are as follows:

```
├── 1.14.4
│   ├── 0001-gluploa d-Only-offer-DMABuf-caps-feature-if-using-EGL.patch
│   ├── 0002-gl-gbm-allow-headless-mode.patch
│   ├── 0003-gst-libs-include-config.h-in-all-source-files.patch
│   ├── 0004-glmemory-Fix-n_wrapped_pointers-usage.patch
│   ├── 0005-gl-egl-Add-gst_egl_image_from_dmabuf_direct-function.patch
│   ├── 0006-gluploa d-try-to-use-the-last-method-after-reconfigur.patch
│   ├── 0007-gluploa d-allow-system-memory-for-dmabuf-in-transform.patch
│   ├── 0008-gluploa d-handle-upload-methods-with-different-caps.patch
│   ├── 0009-gluploadelement-try-to-avoid-dropping-buffers.patch
│   ├── 0010-gluploa d-Implement-direct-dmabuf-uploader.patch
│   ├── 0011-gluploa d-calculate-DRM-fourcc-once-for-direct-dmabuf.patch
│   ├── 0012-gluploa d-debug-output-from-dmabuf-and-dmabuf_direct-.patch
│   ├── 0013-gluploa d-dmabuf-direct-query-formats-before-modifier.patch
│   ├── 0014-gluploa d-dmabuf-direct-report-driver-limitations-to-.patch
│   ├── 0015-gluploa d-Do-prepend-the-preferred-caps.patch
│   ├── 0016-gl-egl-Determine-correct-format-on-dmabuf-import.patch
│   ├── 0017-gluploa d-dmabuf-be-explicit-about-gl-formats-used.patch
│   ├── 0018-opengl-gbm-Adds-missing-unrefs-for-gl-context-and-dr.patch
│   ├── 0019-gst-gl-Remove-duplicate-declarations.patch
│   ├── 0020-gl-gbm-Improve-logging-output.patch
│   ├── 0021-gl-gbm-Add-GST_GL_GBM_DRM_CONNECTOR-environment-vari.patch
│   ├── 0022-gl-window-gbm-Remove-unneeded-extra-function.patch
│   ├── 0023-gl-window-gbm-Remove-unused-private-class-member.patch
│   ├── 0024-gl-window-gbm-Restore-CRTC-on-close.patch
│   ├── 0025-gluploa d-dmabuf-use-out_info-to-create-allocation-pa.patch
│   ├── 0026-gluploa d-Don-t-leak-caps-features.patch
│   ├── 0027-gl-fix-a-few-other-leaks-when-not-getting-to-PAUSED.patch
│   ├── 0028-gl-Don-t-restore-the-viewport-on-function-exit.patch
│   ├── 0029-gluploadelement-Fix-caps-leak.patch
│   ├── 0030-gluploa d-prevent-segfault-when-updating-caps.patch
```

```
│   ├── 0031-glupload-Keep-track-of-cached-EGLImage-texture-forma.patch
│   ├── 0032-eglimage-Fix-memory-leak.patch
│   ├── 0033-glimagesink-fix-upper-left-and-upper-right-rotate-ma.patch
│   ├── 0034-gl-egl-support-direct-dmabuf-import-with-external-oe.patch
│   ├── 0035-gl-gbm-ensure-we-call-the-resize-callback-before-att.patch
│   ├── 0036-glupload-dmabuf-support-direct-upload-into-external-.patch
│   ├── 0037-glupload-fix-transform_caps-NULL-pointer-dereference.patch
│   ├── 0038-glupload-dmabuf-add-DirectDmabufExternal-uploader.patch
│   ├── 0039-glupload-dmabuf-only-accept-uploads-to-external-oes-.patch
│   ├── 0040-video-Add-NV12_10LE40-pixel-format.patch
│   ├── 0041-parsebin-Add-missing-locks-unlocks-of-the-chain-mute.patch
│   ├── 0042-video-Add-gst_video_info_set_interlaced_format.patch
│   ├── 0043-video-format-add-gst_video_format_info_component.patch
│   ├── 0044-video-info-add-gst_video_info_align_full.patch
│   ├── 0045-playbin3-Fix-qt-videoplayer-cannot-change-video-stat.patch
│   ├── 0046-playbin2-Add-preferred-audio-video-sink.patch
│   ├── 0047-HACK-xvimagesink-Support-dma-buffer-rendering.patch
│   ├── 0048-video-converter-Support-rockchip-RGA-2D-accel.patch
│   ├── 0049-HACK-gl-egl-allow-direct-dmabuf-import-when-unable-t.patch
│   ├── 0050-glupload-dmabuf-prefer-DirectDmabufExternal-uploader.patch
│   ├── 0051-videoconvert-Support-preferred-formats.patch
│   ├── 0052-playbin2-Fix-deadlock-when-hooking-about-to-finish-s.patch
│   ├── 0053-HACK-xvimage-Support-NV12_10-and-NV16-dma-buffer.patch

├── 1.18.5
│   ├── 0001-gst-libs-gst-video-gstvideoaggregator.c-fix-build-wi.patch
│   ├── 0002-playbin3-Fix-qt-videoplayer-cannot-change-video-stat.patch
│   ├── 0003-playbin2-Add-preferred-audio-video-sink.patch
│   ├── 0004-HACK-xvimagesink-Support-dma-buffer-rendering.patch
│   ├── 0005-video-converter-Support-rockchip-RGA-2D-accel.patch
│   ├── 0006-HACK-gl-egl-allow-direct-dmabuf-import-when-unable-t.patch
│   ├── 0007-glupload-dmabuf-prefer-DirectDmabufExternal-uploader.patch
│   ├── 0008-videoconvert-Support-preferred-formats.patch
│   ├── 0009-playbin2-Fix-deadlock-when-hooking-about-to-finish-s.patch
│   ├── 0010-HACK-xvimage-Support-NV12_10-and-NV16-dma-buffer.patch
```

5.6 gst-rkmpp

gstreamer-rockchip is an audio and video codec middleware based on gstreamer that adapts to Rockchip platform, and mainly used to connect to the mpp interface.

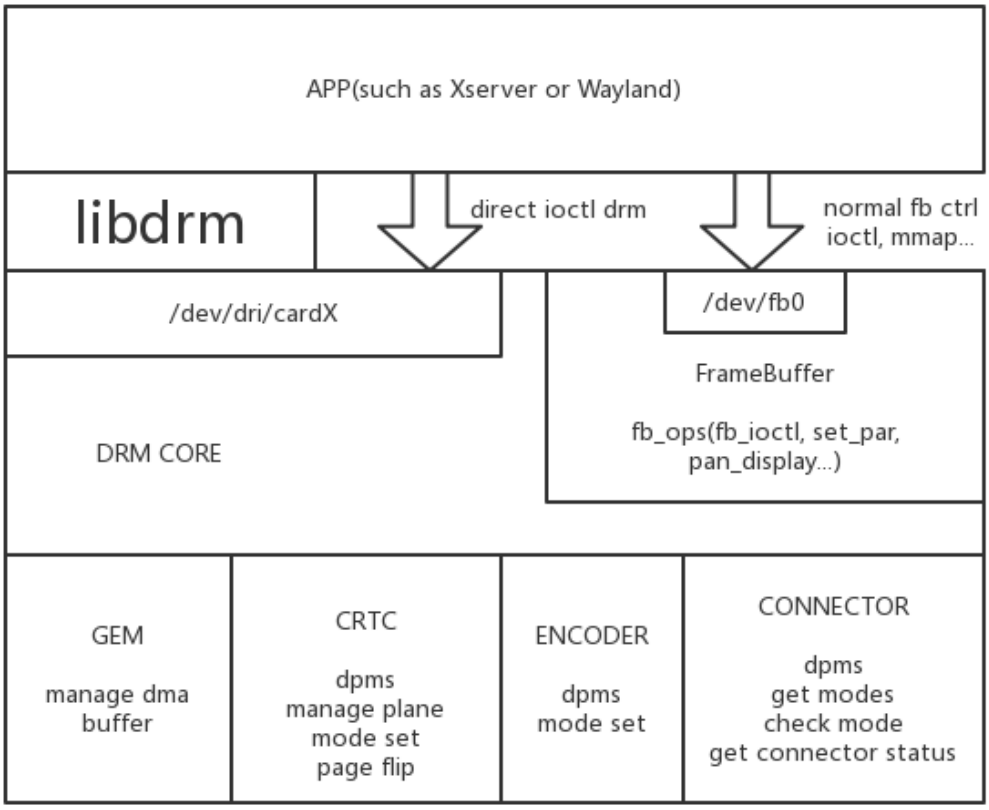
The gstreamer-rockchip pre-built package is as follows:

```
gst-rkmpp/
├── gstreamer1.0-rockchip1-dbgsym_1.14-4_arm64.deb
└── gstreamer1.0-rockchip1_1.14-4_arm64.deb
```

5.7 libdrm

Enable kmssink support based on the official LIBDRM 2.4.97 version.

Libdrm is a cross driver middleware that allows user space applications (for example, as Mesa and 2D drivers) to communicate with kernel through DRI. Please refer to the following DRM structure diagram:



The libdrm pre-built packages are as follows:

```
libdrm
├── libdrm-common_2.4.97-1_all.deb
├── libdrm-dev_2.4.97-1_arm64.deb
├── libdrm2-dbgsym_2.4.97-1_arm64.deb
├── libdrm2_2.4.97-1_arm64.deb
├── libkms1-dbgsym_2.4.97-1_arm64.deb
└── libkms1_2.4.97-1_arm64.deb
```

5.8 libdrm-cursor

There are three key features of this package:

- vop is without a mouse layer, it supports using overlay as a mouse layer
- Support display overlay layer in AFBC format
- handling the abnormality that limit the mouse beyond the boundary

The drm-cursor configuration functions are as follows:

```
cat  /etc/drm-cursor.conf

# Configure file for libdrm-cursor.
#
#debug=
# log-file=
# hide=1 # hide cursors
# atomic=0 # disable atomic drm API
```

```
# max-fps=60
# allow-overlay=1 # allowing overlay planes
# prefer-afbc=0 # prefer plane with AFBC modifier supported
# num-surfaces=8 # num of egl surfaces to avoid edge moving corruption
# prefer-plane=65
# prefer-planes=61,65
# crtc-blocklist=64,83
```

The default log is in /var/log/drm-cursor.log

The libdrm-cursor pre-built packages are as follows:

```
libdrm-cursor/
├── libdrm-cursor-dbgsym_1.2.3-1_arm64.deb
├── libdrm-cursor-dev_1.2.3-1_arm64.deb
└── libdrm-cursor_1.2.3-1_arm64.deb
```

5.9 libmali

ARM provides userspace GPU driver, GPU provides opengles, egl, opencl API.
Rockchip provides a series of Mali pre-built deb packages.

Naming rules: GPU model-Software version-Hardware version (if there is, such as r1p0 to distinguish RK3288 and RK3288w)-build options.
Pay attention to the build options:
It is x11-gbm when without suffix. Note that GBM is the memory mechanism used to configure DRM. Do not use fbdev if it is not a 3.10 kernel. GBM is used by QT EGLFS program, which does not depend on X11, Wayland. Wayland/ Wayland-gbm is used by Wayland.

```
libmali/
├── libmali-bifrost-g31-rxp0-x11_1.9-1_arm64.deb
├── libmali-bifrost-g52-g2p0-dummy-dbgsym_1.9-1_arm64.deb
├── libmali-bifrost-g52-g2p0-dummy-gbm-dbgsym_1.9-1_arm64.deb
├── libmali-bifrost-g52-g2p0-dummy-gbm_1.9-1_arm64.deb
├── libmali-bifrost-g52-g2p0-dummy_1.9-1_arm64.deb
├── libmali-bifrost-g52-g2p0-gbm-dbgsym_1.9-1_arm64.deb
├── libmali-bifrost-g52-g2p0-gbm_1.9-1_arm64.deb
├── libmali-bifrost-g52-g2p0-wayland-dbgsym_1.9-1_arm64.deb
├── libmali-bifrost-g52-g2p0-wayland_1.9-1_arm64.deb
├── libmali-bifrost-g52-g2p0-without-cl-dummy-gbm-dbgsym_1.9-1_arm64.deb
├── libmali-bifrost-g52-g2p0-without-cl-dummy-gbm_1.9-1_arm64.deb
├── libmali-bifrost-g52-g2p0-x11-dbgsym_1.9-1_arm64.deb
├── libmali-bifrost-g52-g2p0-x11_1.9-1_arm64.deb
├── libmali-bifrost-g52-r25p0-dummy-dbgsym_1.9-1_arm64.deb
├── libmali-bifrost-g52-r25p0-dummy_1.9-1_arm64.deb
├── libmali-bifrost-g52-r25p0-wayland-dbgsym_1.9-1_arm64.deb
├── libmali-bifrost-g52-r25p0-wayland_1.9-1_arm64.deb
├── libmali-bifrost-g52-r25p0-x11-dbgsym_1.9-1_arm64.deb
├── libmali-bifrost-g52-r25p0-x11_1.9-1_arm64.deb
...
```

5.10 rga

Rockchip RGA is a independent 2D raster graphics acceleration unit. It speeds up 2D graphics operations, such as point/line drawing, image scaling, rotation, bitmap, image composition, etc.

The pre-built packages are as follows:

```
rga/
├── librga-dev_2.1.0-1_arm64.deb
├── librga2-deb
└── librga2_2.1.0-1_arm64.deb
```

5.11 openbox

Openbox is a window manager, not a desktop environment. Openbox is only responsible for maintaining windows opened on the screen. Added support for window outline movement based on the official v3.6.1 version.

```
Modify /home/linaro/.config/openbox/lxde-rc.xml to change
<drawContents>yes</drawContents>  to <drawContents>no</drawContents>
```

For details, please refer to Openbox

The pre-built packages are as follows:

```
openbox/
└── openbox_3.6.1-8_arm64.deb
```

5.12 pcmanfm

Is a lightweight file manager. Added support for outline based on the official version 1.3.1.

The pre-built packages are as follows:

```
pcmanfm/
├── pcmanfm-dbg_1.3.1-1_arm64.deb
└── pcmanfm_1.3.1-1_arm64.deb
```

5.13 rkaiq

The full name is Rockchip Automatic Image Quality, which is a processor used to automatically adjust the image signal. It is mainly used to implement Camera 3A effect, suitable for ISP2.x chips, such as RK3566, RK3568, RK3588...etc.

The pre-built packages are as follows:

```
rkaiq/
└── camera_engine_rkaiq_v0.1.0_arm64.deb
```

5.14 rkisp

The full name is Rockchip Image Signal Processor, image signal processor. Which is mainly used to implement Camera 3A effect, suitable for ISP1.X chips, such as RK3288, RK33999...etc.

The pre-built packages are as follows:

```
rkisp/
└── rkisp-engine-2.2.0_arm64.deb
```

5.15 rkwifibt

Based on the Rockchip platform, WIFI-BT related modules have been debugged, including Firmware, tools, configuration files, etc.

The pre-built packages are as follows:

```
rkwifibt/
├── rkwifibt-broadcom-firmware_1.0.0-1_arm64.deb
├── rkwifibt-dev-tools-dbgsym_1.0.0-1_arm64.deb
└── rkwifibt-dev-tools_1.0.0-1_arm64.deb
```

5.16 xserver

X server is the abbreviation of graphical interface server in Linux system. Common Linux interface operating environments include KDE and GNOME, and X server provides system support for them. Currently, Debian uses the lightweight LXDE desktop environment, and there are many Linux desktop environments and window managers, the comparison between them is as follows:

Desktop Environment/ Window Manager	Memory Usage	CPU Usage	Type
KDE 4.6	363MB	4%	Desktop Environment
Unity	271MB	14%	Desktop Environment (shell)
GNOME 3	193MB	10%	Desktop Environment
GNOME 2.x	191MB	1%	Desktop Environments
XFCE 4.8	144MB	10%	Desktop Environment
LXDE	85MB	10%	Desktop Environment
IceWM	85MB	2%	Desktop Environment
Enlightenment (E17 Standard)	72MB	1%	Window Manager
Fluxbox	69MB	1%	Window Manager
OpenBox	60MB	1%	Window Manager
JWM	58MB	1%	Window Manager

The Xserver provided by Rockchip has the support of two acceleration modes, glamor and exa. It is mainly configured through the file `/etc/X11/xorg.conf.d/20-modesetting.conf`.

The detailed configuration files are described as follows:

```
root@linaro-alip:~# cat /etc/X11/xorg.conf.d/20-modesetting.conf .
Section "Device"
    Identifier  "Rockchip Graphics"
    Driver      "modesetting"

### Use Rockchip RGA 2D HW accel
#    Option      "AccelMethod"    "exa"

### Use GPU HW accel
    Option      "AccelMethod"    "glamor"

    Option      "DRI"            "2"

### Set to "always" to avoid tearing, could lead to up 50% performance loss
    Option      "FlipFB"         "none"

### Limit flip rate and drop frames for "FlipFB" to reduce performance lost
#    Option      "MaxFlipRate"    "25"

    Option      "NoEDID"         "true"
EndSection

Section "Screen"
    Identifier  "Default Screen"
    Device      "Rockchip Graphics"
    Monitor     "Default Monitor"
EndSection
```

```
### Valid values for rotation are "normal", "left", "right"
Section "Monitor"
    Identifier  "Default Monitor"
    Option      "Rotate" "normal"
EndSection
```

The pre-built packages are as follows:

```
Debian10 xserver/
├── xserver-common_1.20.4-1+deb10u3_all.deb
├── xserver-xorg-core_1.20.4-1+deb10u3_arm64.deb
└── xserver-xorg-legacy_1.20.4-1+deb10u3_arm64.deb

Debian11 xserver/
├── xdmx-dbgsym_1.20.11-1_arm64.deb
├── xdmx-tools-dbgsym_1.20.11-1_arm64.deb
├── xdmx-tools_1.20.11-1_arm64.deb
├── xdmx_1.20.11-1_arm64.deb
├── xnest-dbgsym_1.20.11-1_arm64.deb
├── xnest_1.20.11-1_arm64.deb
├── xorg-server-source_1.20.11-1_all.deb
├── xserver-common_1.20.11-1_all.deb
├── xserver-xephyr-dbgsym_1.20.11-1_arm64.deb
├── xserver-xephyr_1.20.11-1_arm64.deb
├── xserver-xorg-core-dbgsym_1.20.11-1_arm64.deb
├── xserver-xorg-core_1.20.11-1_arm64.deb
├── xserver-xorg-dev_1.20.11-1_arm64.deb
├── xserver-xorg-legacy-dbgsym_1.20.11-1_arm64.deb
├── xserver-xorg-legacy_1.20.11-1_arm64.deb
├── xvfb-dbgsym_1.20.11-1_arm64.deb
├── xvfb_1.20.11-1_arm64.deb
├── xwayland-dbgsym_1.20.11-1_arm64.deb
└── xwayland_1.20.11-1_arm64.deb
```

Boot log in `/var/log/Xorg*`, you can check the version of Xserver by the following way:

```
root@linaro-alip:~# cat /var/log/Xorg.0.log |grep "X.Org X Server"
X.Org X Server 1.20.4
```

The commit corresponding to Rockchip modification can be checked as follows:

```
root@linaro-alip:~# cat /var/log/Xorg.0.log |grep xorg-server
[    26.786] xorg-server f805fe554 modesetting: Filter out invalid format
modifiers (https://www.debian.org/support)
```

6. Basic Abilities of Debian Development

6.1 Rebuild Debian Software Packages

The steps to modify and repackage Debian third-party packages are as follows:

- apt source ##Download source code
- create git, add patch
- apt-get build-dep ##Install building dependencies
- dpkg-buildpackage -b -uc -us -d ## build and package

6.2 Build Debian docker

Currently, the relevant source code can be built through Docker and packaged into deb, which is easy to integrate into the system.

Detailed reference documents：
`<SDK>/docs/Linux/ApplicationNote/Rockchip_Developer_Guide_Debian_Docker_EN.pdf`

6.3 Debian Graphics Adaptation Solution

At present, it mainly supports the display architecture of X11 and Wayland, and the widely used combinations are as follows:

- X11 system default adaptation combination:

```
 xfce4/lxde+xserver+lightdm
```

- Wayland system default adaptation combination:

```
gnome+wayland+gdm3
```

6.3.1 Display Architecture Adaptation Solution

X11/Xserver

Wayland/Weston

6.3.2 Window Management Adaptation Solution

openbox/kwin/xfce

6.3.3 Desktop Environment Adaptation Solution

lightdm/gnome/kde

6.3.4 Chromium Adaptation Solution

6.3.5 Debian Glmark2, Glxgears Adaptation

6.3.6 Debian Boot Logo or Animation Adaptation

6.3.7 Debian Panfrost Adaptation Solution

6.4 Debian Audio and Video Adaptation Solution

Firstly, introduce the general process of video encoding and decoding on Rockchip platforms

```
vpu_service --> mpp --> gstreamer/rockit/rkmedia/ --> app
vpu_service: driver
mpp: video codec middleware for rockchip platform, please refer to mpp document
for related instructions
gstreamer/rockit/rkmedia: components for connecting apps
```

At present, GStreamer is mainly used to connect apps and codec components in Debian systems.

The codec function can also be tested directly through the mpp test interface (such as mpi_dec_test\mpi_enc_test...)
For mpp source code, please refer to SDK/external/mpp/
For test demo, please refer to SDK/external/mpp/test
For more details, please refer to SDK document Rockchip_Developer_Guide_MPP_CN.pdf`

6.4.1 Audio Pulseaudio Channel Adaptation

6.4.2 MPP and VPU Adaptation

6.4.3 GStreamer Adaptation

6.4.4 Rockit Adaptation

6.4.5 QT/MPV/QSplayer/Parole Players Adaptation

6.5 Debian Network Adaptation Solution

6.5.1 RKWIFIBT Adaptation

6.5.2 Ethernet Adaptation

6.5.3 3G/4G/5G Module Adaptation

6.5.4 Network Management Adaptation

6.6 Debian Camera Adaptation Solution

6.6.1 rkisp Adaptation

6.6.2 rkaiq Adaptation

6.6.3 GStreamer/Rockit Channels Adaptation

6.6.4 Structured Light Modules Adaptation

6.7 Debian Power Management Adaptation Solution

6.7.1 Suspend and Resume Adaptation

6.7.2 Battery Charge and Discharge Adaptation

6.7.3 Power on and up Adaptation

6.7.4 Power Key Adaptation

6.8 Debian AI Adaptation Solution

6.8.1 RKNPU Adaptation

6.8.2 RKNN Test Demo Adaptation

6.9 Debian Security Upgrade Adaptation Solution

6.9.1 Secureboot Security Adaptation

6.9.2 Recovery Upgrade Adaptation

6.9.3 OTA Upgrade Adaptation

6.10 Debian Touch Adaptation Solution

6.10.1 Touch Screen Adaptation

6.10.2 Touchpad Adaptation

6.10.3 Mouse Adaptation

6.11 Debian Sensor Adaptation Solution

gsensor/lsensor..

6.12 Debian System Information

6.13 Debian Crop

6.14 Debian Testing

rockchip_test integrates functional, stress, and performance related tests

6.15 Debian Debugging Tools

6.16 Debian Performance Optimization

6.16.1 Performance Optimization

6.16.2 Memory Optimization

6.16.3 Boot Optimization

7. Debian FAQ

7.1 "noexec or nodev" Issue

```
noexec or nodev issue /usr/share/debootstrap/functions: line 1450:
..../rootfs/ubuntu-build-service/buster-desktop-arm64/chroot/test-dev-null:
Permission denied E: Cannot install into target '/rootfs/ubuntu-build-
service/buster-desktop-arm64/chroot' mounted with noexec or nodev
```

Solution:

```
mount -o remount,exec,dev xxx
(The xxx is the project directory path, and then rebuild)
```

In addition, if other compilation exceptions are encountered, first check that the compilation system used is not the system type of ext2/ext4.

7.2 Failed to Download "Base Debian"

- Since building Base Debian needs to visit foreign websites, and when using domestic networks to visit foreign websites, download failures often occur:

Debian uses live build, the mirror source is changed to domestic and can be configured like this:

```
32-bit system:

+++ b/ubuntu-build-service/{buster/bullseye}-desktop-armhf/configure
@@ -11,6 +11,11 @@ set -e
 echo "I: create configuration"
 export LB_BOOTSTRAP_INCLUDE="apt-transport-https gnupg"
 lb config \
+ --mirror-bootstrap "http://mirrors.ustc.edu.cn/debian" \
+ --mirror-chroot "http://mirrors.ustc.edu.cn/debian" \
+ --mirror-chroot-security "http://mirrors.ustc.edu.cn/debian-security" \
+ --mirror-binary "http://mirrors.ustc.edu.cn/debian" \
+ --mirror-binary-security "http://mirrors.ustc.edu.cn/debian-security" \
  --apt-indices false \
  --apt-recommends false \
  --apt-secure false \

64-bit system:
  --- a/ubuntu-build-service/{buster/bullseye}-desktop-arm64/configure
+++ b/ubuntu-build-service/{buster/bullseye}-desktop-arm64/configure
@@ -11,6 +11,11 @@ set -e
 echo "I: create configuration"
 export LB_BOOTSTRAP_INCLUDE="apt-transport-https gnupg"
 lb config \
+ --mirror-bootstrap "http://mirrors.ustc.edu.cn/debian" \
+ --mirror-chroot "http://mirrors.ustc.edu.cn/debian" \
+ --mirror-chroot-security "http://mirrors.ustc.edu.cn/debian-security" \
+ --mirror-binary "http://mirrors.ustc.edu.cn/debian" \
+ --mirror-binary-security "http://mirrors.ustc.edu.cn/debian-security" \
  --apt-indices false \
  --apt-recommends false \
  --apt-secure false \
```

If the package cannot be downloaded due to other network reasons, there is a pre-built package to share in Baidu Cloud Network Disk, place it in the current directory and execute the next step directly .

7.3 Abnormal Operation Causes an error to Mount /dev

For example, like "askpass command or cannot use one" appears

It may be frequent abnormal operations (CTRL+C) during the compilation process, and the above errors can be fixed by the following methods:

```
 sudo -S  umount  /dev
```

7.4 How to Check System Related Information

7.4.1 How to Check the Debian Version of Your System

```
root@linaro-alip:~# cat /etc/debian_version
11.1
```

7.4.2 How to Check Whether the Debian Display Uses X11 or Wayland

On X11 systems:

```
$ echo $XDG_SESSION_TYPE
x11
```

On X11 systems:

```
$ echo $XDG_SESSION_TYPE
wayland
```

7.4.3 How to Check System Partition Status

```
root@linaro-alip:~# parted -l

Model: MMC BJTD4R (sd/mmc)
Disk /dev/mmcblk0: 31.3GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name      Flags
 1      8389kB  12.6MB  4194kB               uboot
 2      12.6MB  16.8MB  4194kB               misc
 3      16.8MB  83.9MB  67.1MB               boot
```

```
4       83.9MB  218MB   134MB                recovery
5       218MB   252MB   33.6MB               backup
6       252MB   15.3GB  15.0GB  ext4         rootfs
7       15.3GB  15.4GB  134MB   ext2         oem
8       15.6GB  31.3GB  15.6GB  ext2         userdata
```

8. Debian Third Party Open Source Software and License Instructions

For more information about Debian open source, please refer to the official website legal

9. Debian Reference Materials

Please refer to the official Debian documentations debian-docs