Rockchip Debian Developer Guide

ID: RK-KF-YF-913

Release Version: V1.7.0

Release Date: 2023-12-20

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

Debian Vesion	Platform	Chipset
12	ARM	RK3588
11	ARM	RK3588、RK3568、RK3566、RK3562、RK3399、RK3288
10	ARM	RK3399PRO、PX30、RK3326、RK3288、RK3328、RK3126C

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
2022-04-12	V1.1.0	Caesar Wang	Update live-build and FAQ chapters.
2022-5-20	V1.2.0	Caesar Wang	Support for Linux5.10
2022-9-20	V1.3.0	Caesar Wang	Update mpp/chromium Update bootaim
2022-11-20	V1.4.0	Caesar Wang	Update partition instruction Update FAQ
2023-04-20	V1.5.0	Caesar Wang	Add blueman/rktoolkit repository Update Camera/NPU/Upgrade Delete some TODO lists
2023-05-20	V1.5.1	Caesar Wang	Add Cheese instructions Add power management instructions
2023-09-20	V1.6.0	Caesar Wang	Update Cheese documentation Add support for Debian 12 Update Debian security and roadmap
2023-12-20	V1.7.0	Caesar Wang	Update Cheese and rkaiq documentation Add Debian FAQ section Optimize some sections of Debian content

Contents

1. Debian Overview

1.1 Overview

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)、Bullseye (11) and Bookworm(12) 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 Debian System Versions Supported

Versions	Supported Architectures	EOL	Current status
Debian 9 “Stretch”	armhf and arm64	2022-06	No longer maintain
Debian 10 “Buster”	armhf and arm64	2024-06	Under maintenance
Debian 11 “Bullseye”	armhf and arm64	2026-06	Under maintenance
Debian 12 “Bookworm”	armhf and arm64	2028-06	Under Development
More Debian Long Term Support Version time, please refer to the official website.			

2. Debian Quick Start Guide

2.1 Setting Up the Environment

We recommend using the Ubuntu 22.04 system for compilation. Other Linux versions may require adjustments to the software packages. In addition to system requirements, there are also specific hardware and software requirements.
Hardware requirements: 64-bit system with more than 40G of disk space. If you are building multiple projects, you will need more disk space.
Software requirements for Ubuntu 22.04 system:
The commands to install the software packages required for setting up the SDK compilation environment are as follows:

```
sudo apt-get update && sudo apt-get install git ssh make gcc libssl-dev \
liblz4-tool expect expect-dev 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 \
libgucharmap-2-90-dev bzip2 expat gpgv2 cpp-aarch64-linux-gnu libgmp-dev \
libmpc-dev bc python-is-python3 python2
```

It is recommended to use the Ubuntu 22.04 system or a higher version for development. If compilation errors occur, install the corresponding software packages based on the error messages.

2.2 Getting the Source Code

Obtain the source code from the Rockchip code server, which is available in the `<SDK>/debian` directory of the project.

2.3 Compilation

- Compiling the Debian required installation packages

```
sudo dpkg -i debian/ubuntu-build-service/packages/*
sudo apt-get install -f

Note:
live-build_*.deb is sourced from
https://mirrors.ustc.edu.cn/debian/pool/main/l/live-build/
debootstrap_*.deb is sourced from
https://mirrors.ustc.edu.cn/debian/pool/main/d/debootstrap/

Or, install the dependencies as follows:
sudo apt-get remove live-build
git clone https://salsa.debian.org/live-team/live-build.git --depth 1 -b
debian/1%20230131
cd live-build
rm -rf manpages/po/
sudo make install -j8
```

- Compiling Debian

Enter the SDK project and compile directly

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

Or enter the Debian/ directory:

```
cd debian/
```

Refer to the readme.md in the current directory.

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 npu/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.

For more introduction, please refer to the official website Live manual.

The source repository of Debian packages (VCS: Git)

live-build-git

The source repository of Debian packages (it is available online)

live-build

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. Introduction to Debian Precompiled Packages

```
packages/
├── arm64/armhf
│   ├── blueman
│   ├── cheese
│   ├── chromium
```

```
│   ├── glmark2
│   ├── gst-plugins-bad1.0
│   ├── gst-plugins-base1.0
│   ├── gst-plugins-good1.0
│   ├── gst-plugins-ugly1.0
│   ├── gstreamer
│   ├── gst-rkmpp
│   ├── libdrm
│   ├── libdrm-cursor
│   ├── libmali
│   ├── libv4l
│   ├── mpp
│   ├── openbox
│   ├── pcmanfm
│   ├── rga
│   ├── rga2
│   ├── rkaiq
│   ├── rkisp
│   ├── rknpu2
│   ├── rktoolkit
│   ├── rkwifibt
│   └── xserver
```

5.1 blueman

Blueman is a GTK+ Bluetooth manager. It provides an efficient way to manage the BlueZ API and simplifies several Bluetooth management tasks, such as:

- Dial-up connections for 3G/EDGE/GPRS
- Connecting/creating Bluetooth networks
- Connecting input devices
- Connecting audio devices
- Sending, receiving, and browsing files via OBEX (Object Exchange)
- Pairing

The precompiled packages include:

```
blueman/
├── blueman_2.3.5-2_arm64.deb
└── blueman-dbgsym_2.3.5-2_arm64.deb

packages-patches/blueman/
├── 0001-Blueman-Fix-DisplayPasskey-for-Keyboard.patch
└── 0002-Don-t-turn-off-the-bt-power.patch
```

Rockchip's modifications to the Blueman package mainly address Bluetooth issues such as keyboard/speaker, sleep/wake-up problems, etc.

5.2 mpp

Rockchip's Media Process Platform (MPP) is a universal media processing software platform suitable for Rockchip chip series. The platform shields application software from chip-specific complex lower-level processing, aiming to eliminate differences between various chips and provide a unified Media Process Interface (MPI).

MPP's features include:

- Video Decoding

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

- Video Encoding

  H.264 / VP8 / MJPEG

- Video Processing

  Video copy, scaling, color space conversion, Deinterlacing

To debug issues, activate more logs using the following switches:

```
export mpi_debug=1
export mpp_debug=1
export h264d_debug=1
export mpp_buffer_debug=2
```

By default, mpp logs are not output to the terminal. To enable, do the following:

```
export mpp_syslog_perror=1
```

Debugging log switches for different kernel versions:

```
4.19/5.10 kernel (Linux 4.19 and above)
$ echo 0x100 > /sys/module/rk_vcodec/parameters/mpp_dev_debug
$ cat /proc/kmsg

4.4 kernel (Linux 4.4)
$ echo 0x100 > /sys/module/rk_vcodec/parameters/debug
$ cat /proc/kmsg
```

These commands output the execution time of kernel single-frame encoding/decoding, useful for performance evaluation or analyzing stuttering/smoothness issues. For example, the log information output in RK3568 kernel 5.10:

```
rk_vcodec: fdf80200.rkvdec:0 session 2269:34 time: 14870 us hw 5430 us
rk_vcodec: fdf80200.rkvdec:0 session 2269:34 time: 13157 us hw 4132 us
rk_vcodec: fdf80200.rkvdec:0 session 2269:34 time: 12976 us hw 4098 us
rk_vcodec: fdf80200.rkvdec:0 session 2269:34 time: 11295 us hw 4070 us
...
```

For more information, refer to the MPP development documentation in the SDK's `<SDK>/docs/Linux/Multimedia` directory.

Precompiled packages integrated into 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
```

5.3 libv4l

A v4l2 plugin for interfacing the Chromium browser with mpp for hardware decoding. Precompiled packages:

```
libv4l/
...
├── libv4l-0_1.22.1-5_arm64.deb
├── libv4l-0-dbgsym_1.22.1-5_arm64.deb
├── libv4l2rds0_1.22.1-5_arm64.deb
├── libv4l2rds0-dbgsym_1.22.1-5_arm64.deb
├── libv4lconvert0_1.22.1-5_arm64.deb
├── libv4lconvert0-dbgsym_1.22.1-5_arm64.deb
├── libv4l-dev_1.22.1-5_arm64.deb
├── libv4l-rkmpp_1.5.1-1.1_arm64.deb
├── libv4l-rkmpp-dbgsym_1.5.1-1.1_arm64.deb
├── v4l-utils_1.22.1-5_arm64.deb
└── v4l-utils-dbgsym_1.22.1-5_arm64.deb
```

The third-party v4l-utils package requires the following patches:

```
├── 0001-libv4l2-Support-mmap-to-libv4l-plugin.patch
├── 0002-libv4l-mplane-Filter-out-multiplane-formats.patch
├── 0003-libv4l-add-V4L2_MEMORY_DMABUF-memory-support.patch
├── 0004-libv4l-mplane-plugin-add-exbuf_ioctl-for-dmabuf.patch
├── 0005-video-max-frame-change-from-32-to-64.patch
├── 0006-Support-builtin-v4l-plugins.patch
├── 0007-libv4l-Disallow-conversion-by-default.patch
```

Then compile libv4l-rkmpp, the libv4l plugin interfacing with mpp:

```
├── libv4l-rkmpp_1.5.1-1.1_arm64.deb
├── libv4l-rkmpp-dbgsym_1.5.1-1.1_arm64.deb
```

5.4 chromium

The Chromium browser integrated into Debian is compiled based on Yocto.

```
chromium/
└── chromium-x11_114.0.5735.198_arm64/armhf.deb
```

Chromium browser supports a variety of video formats, including H264, VP8, VP9, and AV1. Starting from Chromium version 105, it has added support for the H265 video format. Currently, in the Debian system, support for Chromium video hardware decoding has been integrated, achieved through a customized version of Chromium, the V4L2 plugin, and efficient MPP hardware decoding.

Custom modifications to Chromium mainly include:

- Modifying Chromium to enable V4L2 VDA support, along with related patches.
- Adding a V4L2 MPP plugin.

The drawbacks are:
a. Decoding only supports VP8, H264, H265, VP9, and AV1.
b. Encoding only supports VP8 and H264.
c. Requires modifying the Chromium compilation (completing the Yocto full compilation process).
Currently, we provide a Chromium Wayland patch, supported on Yocto, Buildroot, and other systems.

- Chromium Version

root@linaro-alip:~# chromium --version
Chromium 114.0.5735.198 stable

5.5 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 refer to 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
```

5.6 gst-rkmpp

The 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 pre-built packages of gstreamer-rockchip are as follows:

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

5.7 GStreamer

Gstreamer includes the core framework and core components.

The patches for GStreamer 1.22.0 are as follows:

```
├── 0001-filesrc-Fix-stopping-race-in-pull-mode.patch
├── 0002-HACK-gstpad-Add-1-sec-timeout-for-activation.patch
├── 0003-HACK-caps-Consider-dmabuf-subset-of-system-memory.patch
├── 0004-gst-launch-Fix-random-hang-when-EOS.patch
```

5.8 gst-plugins-base1.0

gst-plugins-base is a collection of essential plugins required for GStreamer applications.

Based on the official gst-plugins-base version from GStreamer, support for dma buffer and rga/gpu graphic acceleration adaptation has been added. The patches for gst-plugins-base1.0-1.22.0 are as follows:

```
gst1-plugins-base
├── 0001-Revert-decodebin-only-emit-drained-signal-when-top-c.patch
├── 0002-playbin2-send-one-about-to-finish-per-group.patch
├── 0003-playbin-do-not-drain-on-first-EOS.patch
├── 0004-playbin2-Fix-deadlock-when-hooking-about-to-finish-s.patch
├── 0005-playbin3-Fix-Qt-videoplayer-cannot-change-video-stat.patch
├── 0006-playbin2-Add-preferred-audio-video-sink.patch
├── 0007-HACK-xvimagesink-Support-dma-buffer-rendering.patch
├── 0008-video-converter-Support-rockchip-RGA-2D-accel.patch
├── 0009-HACK-gl-egl-allow-direct-dmabuf-import-when-unable-t.patch
├── 0010-glupload-dmabuf-prefer-DirectDmabufExternal-uploader.patch
├── 0011-videoconvert-Support-preferred-formats.patch
├── 0012-glupload-Support-NV12_10LE40-and-NV12-NV12_10LE40-NV.patch
├── 0013-xvimagesink-Defer-prepare-window-when-getting-zero-w.patch
├── 0014-riff-Fix-bps-caculation-error-for-ADPCM.patch
├── 0015-xvimagesink-Allow-disabling-decorations.patch
├── 0016-xvimagesink-Apply-toplevel-window-s-position.patch
├── 0017-xvimagesink-Ignore-expose-redraw-when-video-not-read.patch
├── 0018-gl-x11-Honor-render-rectangle-for-toplevel-window.patch
├── 0019-gl-wayland-Honor-render-rectangle-for-toplevel-windo.patch
├── 0020-gl-x11-Allow-disabling-decorations-for-toplevel-wind.patch
├── 0021-gst-libs-Support-NV16_10LE40.patch
```

5.9 gst-plugins-bad1.0

Gstreamer plugins of lower quality, which may be moved to the good plugins list upon maturation.

Based on the official gst-plugins-bad version from GStreamer, plugins like kmssink and waylandsink have been added to adapt to some features and fix issues specific to the Rockchip platform.

The patches for gst-plugins-bad1.0-1.22.0 are as follows:

```
gst1-plugins-bad
├── 0001-h265parser-Fix-read-vui-error.patch
├── 0002-interim-fix-vc1-stream-may-memory-leak-when-pending.patch
├── 0003-waylandsink-release-frame-callback-when-finalizing.patch
├── 0004-kmssink-Support-render-rectangle-for-plane.patch
├── 0005-kmssink-Request-window-handle.patch
├── 0006-waylandsink-Support-place-below-above.patch
├── 0007-waylandsink-Enable-changing-window-handle.patch
```

```
├── 0008-kmssink-Support-setting-plane-zpos.patch
├── 0009-waylandsink-Support-setting-toplevel-window-position.patch
├── 0010-HACK-gstmpegvideoparse-Split-every-picture.patch
├── 0011-mpegtsdemux-Create-new-PCR-group-for-big-gap.patch
├── 0012-gstjpegparse-Allow-parsebin-to-use-it-for-autopluggi.patch
├── 0013-waylandsink-Drop-frame-when-window-not-ready.patch
├── 0014-waylandsink-Fix-random-crash.patch
├── 0015-camerabin2-Support-setting-default-filters.patch
├── 0016-waylandsink-Defer-prepare-window-when-getting-zero-w.patch
├── 0017-mpegts-Support-ignoring-broken-PCR-streams-by-defaul.patch
├── 0018-waylandsink-Fix-crash-when-setting-fullscreen-proper.patch
├── 0019-waylandsink-Support-window-layer-property.patch
├── 0020-waylandsink-Support-window-alpha-property.patch
├── 0021-waylandsink-Support-window-fill-mode-property.patch
├── 0022-HACK-kmssink-Open-drm-devnode-directly.patch
├── 0023-waylandsink-Use-create_immed-to-create-dmabuf.patch
├── 0024-waylandsink-Support-frame-sync-mode.patch
├── 0025-kmssink-Support-NV12_10LE40-and-NV12-NV12_10LE40-NV1.patch
├── 0026-waylandsink-Support-NV12_10LE40-and-NV12-NV12_10LE40.patch
├── 0027-waylandsink-Use-the-correct-video-info-to-access-all.patch
├── 0028-waylandsink-Prefer-to-use-waylandsink.patch
├── 0029-kmssink-Avoid-double-closing-shared-gem-handle.patch
├── 0030-kmssink-Support-ignoring-aspect-ratio.patch
├── 0031-kmssink-Support-setting-prefered-frame-syncing-mode.patch
├── 0032-waylandsink-Support-pointer-and-touch.patch
├── 0033-waylandsink-Parse-video-size-in-propose_allocation.patch
├── 0034-waylandsink-Wait-10s-for-toplevel-window-s-configure.patch
├── 0035-waylandsink-Fix-buffer-size-error-when-video-cropped.patch
├── 0036-HACK-waylandsink-Set-size-in-wl_subsurface_set_posit.patch
├── 0037-waylandsink-Support-transparent-video.patch
├── 0038-kmssink-Improve-monitor-and-plane-selection.patch
├── 0039-kmssink-Support-scaling-in-modesetting.patch
├── 0040-kmssink-Support-fullscreen-prop.patch
├── 0041-av1parser-Don-t-consider-unknown-metadata-OBUs-a-bit.patch
├── 0042-kmssink-Avoid-src-size-overflow.patch
├── 0043-waylandsink-Support-force-trying-dmabuf.patch
├── 0044-kmssink-Support-NV16_10LE40.patch
```

5.10 gst-plugins-good.0

Gstreamer's high-quality plugins are licensed under LGPL.

Based on the official GStreamer version of gst-plugins-good, additional plugins such as v4l2 and rga have been added to adapt to some features of the Rockchip platform and fix issues.

The patches for gst-plugins-good1.0-1.22.0 are as follows:

```
gst1-plugins-good/
├── 0001-qtdemux-don-t-skip-the-stream-duration-longer-than-3.patch
├── 0002-HACK-flacparse-Handle-metadata-127.patch
├── 0003-autodetect-Add-preferred-for-autovideosink-autoaudio.patch
├── 0004-v4l2-Support-preferred-formats.patch
├── 0005-video-flip-Support-rockchip-RGA-2D-accel.patch
├── 0006-v4l2src-Support-setting-default-device.patch
├── 0007-v4l2src-Filter-out-unavailable-RK-sources.patch
├── 0008-v4l2src-Support-setting-max-resolution.patch
├── 0009-matroska-demux-parse-Disable-QoS-lace-skipping-by-de.patch
├── 0010-v4l2-Increase-VIDEO_MAX_FRAME-to-64.patch
├── 0011-v4l2-Support-setting-v4l2-min-buffers.patch
├── 0012-v4l2-Support-disabling-buffer-sharing.patch
```

5.11 libdrm

Enable kmssink support based on the official LIBDRM 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 pre-built packages of libdrm are as follows:

```
libdrm
├── libdrm2_2.4.104-1_arm64.deb
├── libdrm-common_2.4.104-1_all.deb
├── libdrm-dev_2.4.104-1_arm64.deb
├── libkms1_2.4.104-1_arm64.deb
└── libkms1-dbgsym_2.4.104-1_arm64.deb
```

Patches as below:

```
├── 0001-tests-meson.build-disable-nouveau-tests-for-static-b.patch
├── 0002-modetest-Speed-up-dumping-info.patch
├── 0003-modetest-Fix-set-property-error-when-using-atomic.patch
├── 0004-HACK-Open-rockchip-drm-device-by-default.patch
├── 0005-HACK-Bypass-drm-GetBusid-drmAuthMagic-SetInterfaceVe.patch
```

5.12 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

Configuration function of drm-cursor 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 pre-built packages of libdrm-cursor are as follows:

```
libdrm-cursor/
├── libdrm-cursor_1.3.0-1_arm64.deb
├── libdrm-cursor-dbgsym_1.3.0-1_arm64.deb
└── libdrm-cursor-dev_1.3.0-1_arm64.deb
```

5.13 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-valhall-g610-g6p0-dummy_1.9-1_arm64.deb
├── libmali-valhall-g610-g6p0-dummy-dbgsym_1.9-1_arm64.deb
├── libmali-valhall-g610-g6p0-dummy-gbm_1.9-1_arm64.deb
├── libmali-valhall-g610-g6p0-dummy-gbm-dbgsym_1.9-1_arm64.deb
├── libmali-valhall-g610-g6p0-gbm_1.9-1_arm64.deb
├── libmali-valhall-g610-g6p0-gbm-dbgsym_1.9-1_arm64.deb
├── libmali-valhall-g610-g6p0-wayland_1.9-1_arm64.deb
├── libmali-valhall-g610-g6p0-wayland-dbgsym_1.9-1_arm64.deb
├── libmali-valhall-g610-g6p0-without-cl-dummy-gbm_1.9-1_arm64.deb
├── libmali-valhall-g610-g6p0-without-cl-dummy-gbm-dbgsym_1.9-1_arm64.deb
├── libmali-valhall-g610-g6p0-x11_1.9-1_arm64.deb
└── libmali-valhall-g610-g6p0-x11-dbgsym_1.9-1_arm64.deb
...
```

5.14 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

rga2
├── librga2_2.2.0-1_arm64.deb
├── librga2-dbgsym_2.2.0-1_arm64.deb
└── librga-dev_2.2.0-1_arm64.deb
```

5.15 openbox

Openbox is a window manager, not a desktop environment. Openbox is only responsible for maintaining windows opened on the screen. Window outline movement support is added 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_3.6.1-9+deb11u1_arm64.deb
```

5.16 pcmanfm

Is a lightweight file manager. Outline supported is add based on the official version 1.3.1.

The pre-built packages are as follows:

```
pcmanfm/
├── pcmanfm_1.3.2-1_arm64.deb
└── pcmanfm-dbgsym_1.3.2-1_arm64.deb
```

5.17 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_rk3568_arm64.deb  ##It is applicable to RK3566/RK3568
chipset
└── camera_engine_rkaiq_rk3588_arm64.deb  ##It is applicable to RK3588 chipset
```

5.18 rkisp

The full name is Rockchip Image Signal Processor, image signal processor. Which is mainly used to implement camera's 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
```

5.19 rkwifibt

WIFI-BT modules which have been debugged based on the Rockchip platform, include 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
```

Rockchip changes the Bluetooth package mainly to fix som Bluetooth issues such as keyboards/speakers.

5.20 rktoolkit

Rktoolkit is a specialized toolkit for Rockchip, such as' io/update/vendor '_ Storage, etc.:
The pre compiled package is as follows:

```
rktoolkit/
├── rktoolkit_ 1.0.0-1_ arm64.deb
└── rktoolkit-dbgsym_ 1.0.0-1_ arm64.deb
```

5.21 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 comparisons between them are as follows:

Desktop Environment/ Window Manager	Memory Usage	CPU Usage	Type
KDE 4.6	363MB	4%	Desktop Environment
Unity	271MB	14%	Desktop Environment
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 supported for two acceleration modes, glamor and exa. They are mainly configured through the file `/etc/X11/xorg.conf.d/20-modesetting.conf`.

Glamor uses GPU for display rendering, and exa uses RGA for dispaly rendering.

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"         "always"

### Limit flip rate and drop frames for "FlipFB" to reduce performance lost
#    Option      "MaxFlipRate"    "60"

    Option      "NoEDID"         "true"
    Option  "UseGammaLUT"    "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
└── xserver-xorg-dev_1.20.4-1+deb10u3_arm64.deb

Debian11 xserver/
├── xserver-common_1.20.11-1_all.deb
├── xserver-xorg-core_1.20.11-1_arm64.deb
├── xserver-xorg-dev_1.20.11-1_arm64.deb
└── xserver-xorg-legacy_1.20.11-1_arm64.deb
```

Boot log is 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 1.20.11
```

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

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

- `apt-get build-dep <pkg>` ##Install compilation dependent packages
- `apt source <pkg>` ##Download the source code of `<pkg>` package
- create git, add patches
- `dpkg-buildpackage -b -uc -us -d` ## repackage and build

For example, change Xserver software package, modify and repackage deb as follows:

```
# Pay attention to confirmation the deb-src is enabled in
`/etc/apt/sources.list`, and update the source code of apt

# Install dependent packages
```

```
`apt-get build-dep xorg-server-source`

# Downdload Xorg-xserver source code

- `apt source xorg-server-source`

# Create git, add patches
$ cd xorg-server_*
$ git init && git add .
$ git commit -s "xxxxx"

# Begin to build and package deb
dpkg-buildpackage -b -uc -us
```

6.2 Build Debian Docker

Currently, it is supported to build Docker and build related source code through PC, and package into a deb which is easy to integrate into the system.

Please refer to the document for details：
`<SDK>/docs/cn/Linux/ApplicationNote/Rockchip_Developer_Guide_Debian_Docker_EN.pdf`

6.3 Debian Partition Management

At present, resize-all.service related services are added to support resizing and other functions of partitions in various formats.
System services are available in `/lib/systemd/system/resize-all.service` and the main execution file are `/usr/bin/resize-helper`
The debug log is in `/tmp/resize-all.log` as follows:

```
root@linaro-alip:~# cat /tmp/resize-all.log
Will now resize all partitions in /proc/mounts
Handling /dev/root / ext4 rw,relatime
Resizing /dev/mmcblk2p6(ext4)
resize2fs 1.46.2 (28-Feb-2021)
Filesystem at /dev/mmcblk2p6 is mounted on /; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/mmcblk2p6 is now 1572864 (4k) blocks long.

Handling devtmpfs /dev devtmpfs
rw,relatime,size=1993072k,nr_inodes=498268,mode=755
Handling proc /proc proc rw,relatime
Handling sysfs /sys sysfs rw,relatime
Handling securityfs /sys/kernel/security securityfs
rw,nosuid,nodev,noexec,relatime
Handling tmpfs /dev/shm tmpfs rw,nosuid,nodev,noexec
Handling devpts /dev/pts devpts rw,relatime,gid=5,mode=620,ptmxmode=666
Handling tmpfs /run tmpfs rw,nosuid,nodev,size=801828k,nr_inodes=819200,mode=755
Handling tmpfs /run/lock tmpfs rw,nosuid,nodev,noexec,relatime,size=5120k
Handling cgroup2 /sys/fs/cgroup cgroup2
rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot
Handling pstore /sys/fs/pstore pstore rw,relatime
Handling configfs /sys/kernel/config configfs rw,relatime
```

```
Handling debugfs /sys/kernel/debug debugfs rw,relatime
Handling tracefs /sys/kernel/tracing tracefs rw,nosuid,nodev,noexec,relatime
Handling fusectl /sys/fs/fuse/connections fusectl rw,nosuid,nodev,noexec,relatime
Handling tmpfs /tmp tmpfs rw,relatime
Handling /dev/mmcblk2p7 /oem ext4 rw,relatime
Handling /dev/mmcblk2p8 /userdata ext4 rw,relatime
Handling adb /dev/usb-ffs/adb functionfs rw,relatime
```

6.4 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
```

6.4.1 Display Architecture Adaptation Solution

- X11/Xserver is currently configured with LXDE/XFCE lightweight desktop environment, and the desktop manager uses lightdm
- WAYLAND is currently configured with GNOME mainstream desktop environment, and the desktop manager uses gdm3

6.4.2 Window Management Adaptation Solution

- Openbox

Openbox is the standard window manager, with fast, lightweight and extensible features. Openbox 3 series is a completely new window manager, which does not inherit the code of any previous similar software, although it still looks like Blackbox (the code of Openbox 2 series is based on Blackbox 0.65.0). Openbox can be used as an independent operating environment, and it can also be used only as a window manager to replace the default window manager of desktop environments such as KDE and Gnome.

- KWin

KWin is a window manager in the X Window system, which is a part of K Desktop Environment (KDE), although it can be used independently Or for other desktop environments. In KDE 4, KWin added support for combination mode and OpenGL.

- Xfwm

The original window manager of Xfwm4 is Xfce4, which uses a simple and flexible pixmap-based theme engine, using images in the .xpm format. Xfwm4 also uses a text file to configure other options. In addition, you can also choose gtk theme colors, which will be described in the gtk color section of the document.

6.4.3 Desktop Environment Adaptation Solution

- GNOME

As a new version of GNOME, GNOME 3 has obvious changes compared with GNOME 2.x. GNOME 3 is relatively intuitive. You can click "Activity" in the upper left corner or swipe to view applications, work partitions, etc.

Currently, Debian or other distributions use GNOME 3 as the default desktop environment.

- KDE

KDE has a gorgeous Windows-like "Start Menu" interface. Many distributions such as OpenSUSE, PCLinuxOS and Mandriva use KDE as the default desktop environment.

- XFCE

With fewer resources occupation than GNOME and KDE, XFCE is suitable for lightweight desktops and similar to the windows interface environment.

- LXDE

LXDE is also one of the four major desktop environments, a lightweight desktop that takes up less resources.

6.4.4 Chromium Adaptation Solution

6.4.4.1 Introduction

The Chromium browser supports a variety of video formats, including H264, VP8, VP9, AV1, etc. Starting from version 105, Chromium has added support for the H265 video format. Currently, in the Debian system, support for Chromium video hardware decoding has been integrated, achieved through a custom version of Chromium, the V4L2 plugin, and efficient MPP hardware decoding.

Custom modifications to Chromium mainly include:

- Modifying Chromium to enable V4L2 VDA support, along with related patches.
- Adding a V4L2 MPP plugin.

The drawbacks are:
a. Decoding only supports VP8, H264, H265, VP9, and AV1.
b. Encoding only supports VP8 and H264.
c. Requires modifying the Chromium compilation (completing the Yocto full compilation process).
Currently, we provide a Chromium Wayland patch, supported on Yocto, Buildroot, and other systems.

The general process is that chromium enables V4L2 VDA/VEA, boots and creates a virtual v4l2 node, and operation of chromium on the virtual device will be intercepted by v4l-utils to v4l-rkmpp plug-in, and change to call mpp interface.

6.4.4.2 Version

root@linaro-alip:~# chromium --version
Chromium 114.0.5735.198 stable

6.4.4.3 How to Test

Test with the following command:

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

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

```
root@linaro-alip:~# /rockchip-test/chromium/test_chromium_with_video.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 hard solution

```
export mpp_syslog_perror=1
echo 0x100 > /sys/module/rk_vcodec/parameters/mpp_dev_debug
```

Linux4.4 can use the following command to check whether to call the hard solution

```
export mpp_syslog_perror=1
echo 0x4 > /sys/module/rk_vcodec/parameters/debug
```

6.4.4.4 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
```

You can check GPU usage to see if hard acceleration is useful:

```
cat /sys/devices/platform/*gpu/utilisation
```

If the acceleration or hard solution is useless, then analyze the GPU-related issues or lib4l adaptation issues or some permissions related issues.

For example, create related device nodes in `/etc/init.d/rockchip.sh`

```
# Create dummy video node for chromium V4L2 VDA/VEA with rkmpp plugin
echo dec > /dev/video-dec0
echo enc > /dev/video-enc0
chmod 660 /dev/video-*
chown root.video /dev/video-*

# The chromium using fixed pathes for libv4l2.so
ln -rsf /usr/lib/*/libv4l2.so /usr/lib/
[ -e /usr/lib/aarch64-linux-gnu/ ] && ln -Tsf lib /usr/lib64
```

In addition, handle the permissions of related kernel nodes of `/etc/udev/rules.d/99-rockchip-permissions.rules` :

```
ACTION=="remove", GOTO="permissions_end"

# VPU devices
KERNEL=="avsd", MODE="0660", GROUP="video"
KERNEL=="vepu", MODE="0660", GROUP="video"
KERNEL=="h265e", MODE="0660", GROUP="video"
KERNEL=="rkvdec", MODE="0660", GROUP="video"
KERNEL=="rkvenc", MODE="0660", GROUP="video"
KERNEL=="mpp_service", MODE="0660", GROUP="video"
KERNEL=="vpu[_-]service", MODE="0660", GROUP="video"
KERNEL=="hevc[_-]service", MODE="0660", GROUP="video"

# RGA device
KERNEL=="rga", MODE="0660", GROUP="video"

# MALI devices (/dev/mali for mali400)
KERNEL=="mali*", MODE="0660", GROUP="video"
```

6.4.4.5 Performance Testing

Display is in GPU acceleration mode by default. The general process of video hardware solution is to enable V4L2 VDA/VEA in chromium, create a virtual v4l2 node when booting, and the operation of chromium on the virtual device will be intercepted by v4l-utils to the v4l-rkmpp plug-in, and change to call mpp interface. Commonly used webpage test browsers are as follows:

- ARES-6

ARES-6 measures the execution time of the latest JavaScript features, and browsers that start quickly and run smoothly have more advantages.

- Basemark Web 3.0

Basemark Web 3.0 is a comprehensive web browser performance benchmark that tests the ability of browser to run web applications, measuring real-world client-side performance to test the bottleneck of browser.

- JetStream 2

JetStream 2 is a JavaScript and WebAssembly benchmark suite focused on most advanced web applications. Browsers that boot quickly, execute code quickly, and run smoothly get higher scores.

- MotionMark 1.2

MotionMark is a graphics benchmark that measures a browser’s capability to animate complex scenes at a target frame rate.

- Octane

Octane, an early general JavaScript performance benchmark, has been retired. It can indeed measure the performance of JS engine, but it cannot truly reflect the optimization of JS engine for web applications in real-world. Optimizations made for Octane often have little impact on real-world web pages, and in some cases, these optimizations can slow down real-world websites.

- Speedometer 2.1

Speedometer is a browser benchmark that measures the responsiveness of web applications. It uses a demo web application to simulate user actions, such as adding a to-do item. Compared with Octane, Speedometer reflect the optimization of JS engine for real-world web applications more truly.

6.4.4.6 Obtaining Other Information

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

```
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/
```

6.4.5 Debian Boot Logo or Animation Adaptation

Currently, Xserver/Weston has added support for boot animation.
The system service is in `/lib/systemd/system/bootanim.service`, and the main execution file is `/usr/bin/bootanim`

For example, create a file that required to be played in `/etc/bootanim.d/`, and debug log is in `/tmp/bootanim.log`

```
/etc/bootanim.d/gst-bootanim.sh

#!/bin/sh
```

```
gst-play-1.0 /etc/bootanim.d/bootanim.mp4 -q --no-interactive

commit 90103840728382ae5e950650ffdf1197e4985974
Author: Caesar Wang <wxt@rock-chips.com>
Date:   Thu Oct 20 15:37:38 2022 +0800

    overlay: add bootanim

    Create /etc/bootanim.d/gst-test.sh
        gst-launch-1.0 videotestsrc ! kmssink &>/dev/null

    Signed-off-by: Caesar Wang <wxt@rock-chips.com>
    Change-Id: I90761eb5ba4b90c04793a4cc959165f6b70c01b0
```

6.4.6 Debian Panfrost Adaptation Solution

Please refer to Debian official website Panfrost Adaptation

6.5 Debian Audio and Video Adaptation Solution

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

```
vpu_service  --> mpp --> GStreamer/rockit --> app
vpu_service: driver
mpp: video codec middleware for rockchip platform, please refer to mpp document
for related introduction
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”

6.5.1 Audio Pulseaudio Channel Adaptation

The default audio uses pulseaudio, Generally, you only need to configure `/etc/pulse/default.pa`

Currently, the SDK is compatible with two Codecs, ES8388 and RK809.

```
+set-default-source alsa_input.platform-es8388-
sound.HiFi__hw_rockchipes8388__source
+set-default-sink alsa_output.platform-es8388-sound.HiFi__hw_rockchipes8388__sink
+set-default-source alsa_input.platform-rk809-
sound.HiFi__hw_rockchiprk809__source
+set-default-sink alsa_output.platform-rk809-sound.HiFi__hw_rockchiprk809__sink
```

If you need to add more Codec support, get relevant information through the following commands.

```
pactl list sinks short
pactl list sources short
```

For details, refer to Debian official website Pulseaduio and
`<SDK>/docs/Common/AUDIO/Rockchip_Developer_Guide_PulseAudio_CN.pdf`

6.5.2 MPP and VPU Adaptation

The default MPP is pre-compiled into deb and integrated in

```
/usr/lib/aarch64-linux-gnu/librockchip_mpp.so
/usr/lib/aarch64-linux-gnu/librockchip_mpp.so.0
/usr/lib/aarch64-linux-gnu/librockchip_mpp.so.1
```

At the same time, make sure that the kernel has `/dev/mpp_service` related nodes.

Please refer to the document:
`<SDK>/docs/Linux/Multimedia/Rockchip_Developer_Guide_MPP_CN.pdf` for details.

6.5.3 GStreamer Adaptation

Please refer to the document:
`<SDK>/docs/Linux/Multimedia/Rockchip_User_Guide_Linux_Gstreamer_CN.pdf` for details.

6.5.4 Rockit Adaptation

Please refer to the document:
`<SDK>/docs/Linux/Multimedia/Rockchip_User_Guide_Linux_Rockit_CN.pdf` for details.

6.6 Debian Network Adapter Solution

6.6.1 RKWIFIBT Adapter

Refer to the specific documentation
`<SDK>/docs/Linux/Wifibt/Rockchip_Developer_Guide_Linux_WIFI_BT_CN.pdf` for details.

6.7 Debian Camera Adapter Solution

The default webcam application used in the system is Cheese, which utilizes `GStreamer->mpp->vpu` for encoding and decoding on the Rockchip platform. If Cheese is not needed, Gstreamer is also integrated to directly call USB cameras and adapt Rockchip's 3A effect for mipi/csi cameras. For example, the test scripts are as follows:

```
/rockchip-test$ tree -L 1 camera/
camera/
├── camera_rkaiq_test.sh
├── camera_rkisp_test.sh
├── camera_stresstest.sh
├── camera_test.sh
└── camera_usb_test.sh
```

6.7.1 Cheese Adaptation

Cheese is a webcam application used in Debian systems, offering interesting features like applying real-time effects, taking photos, and recording videos. It supports almost all common types of cameras, including built-in and USB cameras.

Some features of Cheese include:

- Real-time effects: Cheese allows you to apply real-time effects while taking photos or recording videos. These effects include black and white, cartoon, fisheye, oil painting, etc.
- Photo and video recording: Cheese enables you to take photos and record videos with the camera, supporting various resolutions and video formats.
- Automatic saving: Cheese can automatically save your photos and videos. You can also manually select the save location and file name.
- Video preview: Cheese provides real-time video preview, convenient for previewing effects before shooting.
- Compatibility: Cheese supports almost all common types of cameras, including built-in and USB cameras.

To adapt the Cheese app, handle the following related issues.

6.7.1.1 Cheese Unable to Obtain Valid Device Node

Cheese, by default, searches for valid devices with the video* name. The actual device name information for Rockchip ISP is as follows (including rkisp_mainpath, rkisp_selfpath, etc.).

```
root@rk3568:~# grep '' /sys/class/video4linux/video*/name
/sys/class/video4linux/video0/name:rkisp_mainpath
/sys/class/video4linux/video1/name:rkisp_selfpath
/sys/class/video4linux/video2/name:rkisp_rawwr0
/sys/class/video4linux/video3/name:rkisp_rawwr2
/sys/class/video4linux/video4/name:rkisp_rawwr3
/sys/class/video4linux/video5/name:rkisp_rawrd0_m
/sys/class/video4linux/video6/name:rkisp_rawrd2_s
/sys/class/video4linux/video7/name:rkisp-statistics
/sys/class/video4linux/video8/name:rkisp-input-param
```

Currently resolved through Gstreamer v4l2 plugin filtering.

Set related environment variables in `/etc/profile.d/gst.sh`:

```
# Default device for v4l2src
export GST_V4L2SRC_DEFAULT_DEVICE=/dev/video-camera0

# Available RK devices for v4l2src
export GST_V4L2SRC_RK_DEVICES=_mainpath_selfpath:_bypass:_scale
```

6.7.1.2 Cheese Frame Drop Issue with High-Resolution Cameras

When opening the imx415 with Cheese, the mipi sensor drops frames, as seen below.

```
root@linaro-alip:/# cheese -d rkisp_mainpath

[  488.510646] rkisp0-vir0: MIPI drop frame
[  488.544222] rkisp0-vir0: MIPI drop frame
[  488.577718] rkisp0-vir0: MIPI drop frame
[  488.611408] rkisp0-vir0: MIPI drop frame
[  488.644975] rkisp0-vir0: MIPI drop frame
[  488.678581] rkisp0-vir0: MIPI drop frame

Output     rkisp_mainpath Format:UYVY Size:4672x3504
```

The device's obtained size is 4672x3504, exceeding the hardware support range.

Currently resolved by limiting ISP's mp maximum output to 3840x2160 through the Gstreamer v4l2 plugin.

Set related environment variables in `/etc/profile.d/gst.sh`:

```
# Max resolution for v4l2src
export GST_V4L2SRC_MAX_RESOLUTION=3840x2160
```

6.7.1.3 Cheese Abnormal Format Retrieval

Some formats may be unsupported in the ISP driver, for example, although the kernel ISP driver reports supporting certain formats, many might actually be unsupported.

For instance:

```
root@linaro-alip:~# v4l2-ctl --list-formats-ext -d /dev/video8
ioctl: VIDIOC_ENUM_FMT
        Type: Video Capture Multiplanar

        [0]: 'UYVY' (UYVY 4:2:2)
                Size: Stepwise 32x16 - 4672x3504 with step 8/8
        [1]: '422P' (Planar YUV 4:2:2)
                Size: Stepwise 32x16 - 4672x3504 with step 8/8
        [2]: 'NV16' (Y/CbCr 4:2:2)
                Size: Stepwise 32x16 - 4672x3504 with step 8/8
...
```

Currently, formats like NV12 and N16 are normal, but I420 shows abnormal content. There might be many unsupported formats. Early Windows applications usually prefer RGB formats, and if the ISP driver reports supporting RGB formats but actually doesn't, it can cause many applications to malfunction.

Solution: Currently, the preferred formats are selected through the Gstreamer v4l2 environment variable.

Set related environment variables in `/etc/profile.d/gst.sh`:

```
# Preferred formats for videoconvert
export GST_VIDEO_CONVERT_PREFERRED_FORMAT=NV12:NV16:I420:YUY2
```

6.7.1.4 Cheese Recording Freezing Issue

In Cheese, the hardware encoding uses the software vp8enc, but it causes issues with native pulseaudio and leads to freezing.

Solution: Currently, support for software vp8enc encoding is added through GStreamer and updating pulseaudio

Set the related environment variable in `/etc/profile.d/gst.sh`:

```
export GST_MPP_VP8ENC_FAKE_VP8ENC=1
```

6.7.1.5 Cheese Recording Using mpp Encoding

By default, Cheese uses vvp8enc software encoding, which has poor performance. The default modification is as follows:
Force the use of rockchip mpp's mppvp8enc for hardware encoding

```
--- a/libcheese/cheese-camera.c
+++ b/libcheese/cheese-camera.c
@@ -442,7 +442,7 @@ cheese_camera_set_video_recording (CheeseCamera *camera,
GError **error)
   gboolean res;
   /* Check if we can use global preset for vp8enc. */
-  video_enc = gst_element_factory_make ("vp8enc", "vp8enc");
+  video_enc = gst_element_factory_make ("mppvp8enc", "vp8enc");
   video_preset = (gchar *) &CHEESE_VIDEO_ENC_PRESET;
   res = gst_preset_load_preset (GST_PRESET (video_enc), video_preset);
```

You can confirm it like this:

```
root@linaro-alip:/# gst-inspect-1.0 |grep vp8
vpx:  vp8dec: On2 VP8 Decoder
vpx:  vp8enc: On2 VP8 Encoder
rtp:  rtpvp8depay: RTP VP8 depayloader
rtp:  rtpvp8pay: RTP VP8 payloader
rockchipmpp:  mppvp8enc: Rockchip Mpp VP8 Encoder
```

Below is the encoding support situation for Rockchip chips.

- **Encoding Capability Specification Table**

Chip Name	H264	H265	VP8
RK3588	7680x4320@30f	7680x4320@30f	1920x1088@30f
RK3566/RK3568	1920x1088@60f	1920x1088@60f	N/A
RK3562	1920x1088@60f	N/A	N/A
RK3399	1920x1088@30f	N/A	1920x1088@30f
RK3328	1920x1088@30f	1920x1088@30f	1920x1088@30f
RK3288	1920x1088@30f	N/A	1920x1088@30f
RK3326	1920x1088@30f	N/A	1920x1088@30f
PX30	1920x1088@30f	N/A	1920x1088@30f
RK312X	1920x1088@30f	N/A	1920x1088@30f

6.7.1.6 How to Change Cheese Image and Video Storage Path

The default image or video storage path is in the `~/.gnome2/cheese` directory. To change the path, modify as follows:

```
cat << EOF >~/.config/user-dirs.dirs
> #!/bin/bash
> XDG_DESKTOP_DIR="$HOME/Desktop"
> XDG_DOWNLOAD_DIR="$HOME/Downloads"
> XDG_TEMPLATES_DIR="$HOME/Templates"
> XDG_PUBLICSHARE_DIR="$HOME/Public"
> XDG_DOCUMENTS_DIR="$HOME/Documents"
> XDG_MUSIC_DIR="$HOME/Music"
> XDG_PICTURES_DIR="$HOME/Pictures"
> XDG_VIDEOS_DIR="$HOME/Videos"
> EOF
```

Where `XDG_VIDEOS_DIR` is the video path, and `XDG_PICTURES_DIR` is the image path.

6.7.2 rkisp Adaptation

Supports ISP1.X mainly for chips like RK3399/RK3288/PX30/RK3326 etc.

```
rkisp/
└── camera_engine_rkisp-v2.2.0_arm64.deb
```

6.7.3 rkaiq Adaptation

Supports different versions of ISP such as ISP21/30/32-Lite, mainly for chips like RK3566/RK3568/RK3588/RK3562 etc.

```
rkaiq/
├── camera_engine_rkaiq_rk3562_arm64.deb
├── camera_engine_rkaiq_rk3568_arm64.deb
└── camera_engine_rkaiq_rk3588_arm64.deb
```

6.7.3.1 RKAIQ Debugging

- Confirm rkaiq version and related log

```
root@linaro-alip:/# cat /var/log/syslog |grep rkaiq
...
Jun 18 17:37:50 linaro-alip rkaiq: ************************* VERSION INFOS
*************************
Jun 18 17:37:50 linaro-alip rkaiq: version release date: 2023-10-07
Jun 18 17:37:50 linaro-alip rkaiq:          AIQ:       AIQ v5.0x4.0
Jun 18 17:37:50 linaro-alip rkaiq:    IQ PARSER:       Calib
v1.4.8,magicCode:1170944
Jun 18 17:37:50 linaro-alip rkaiq: ********************** VERSION INFOS END
**********************
```

- Is the rkaiq service running

root@linaro-alip:/# ps aux |grep rkaiq
root 2339 0.0 0.4 163020 8076 ? Sl 17:37 0:00 /usr/bin/rkaiq_3A_server
root 2340 0.0 0.0 8508 1240 ? S 17:37 0:00 logger -t rkaiq
root 2699 0.0 0.0 6264 604 ttyFIQ0 S+ 17:42 0:00 grep rkaiq

- Kernel ISP debugging debug log

echo 3 > /sys/module/video_rkisp/parameters/debug
echo 3 > /sys/module/video_rkcif/parameters/debug

- rkaiq debug log

AE module log:
echo 0x1ff4 > /tmp/.rkaiq_log

AWB module log:
echo 0x2ff4 > /tmp/.rkaiq_log

AF module log:
echo 0x4ff4 > /tmp/.rkaiq_log

HDR module log:
echo 0x20ff3 > /tmp/.rkaiq_log

NR module log:
echo 0x40ff4 > /tmp/.rkaiq_log

Dehaze log:
echo 0x2000ff3 > /tmp/.rkaiq_log

Sharp log:
echo 0x80000ff4 > /tmp/.rkaiq_log

CAMHW log:
echo 0x4000000ff4 > /tmp/.rkaiq_log

- Check device node information

```
root@linaro-alip:/# grep '' /sys/class/video4linux/video*/name
/sys/class/video4linux/video0/name:stream_cif_mipi_id0
/sys/class/video4linux/video1/name:stream_cif_mipi_id1
/sys/class/video4linux/video2/name:stream_cif_mipi_id2
/sys/class/video4linux/video3/name:stream_cif_mipi_id3
/sys/class/video4linux/video4/name:rkcif_scale_ch0
/sys/class/video4linux/video5/name:rkcif_scale_ch1
/sys/class/video4linux/video6/name:rkcif_scale_ch2
/sys/class/video4linux/video7/name:rkcif_scale_ch3
/sys/class/video4linux/video8/name:rkcif_tools_id0
/sys/class/video4linux/video9/name:rkcif_tools_id1
/sys/class/video4linux/video10/name:rkcif_tools_id2
...
```

6.8 Debian Power Management Adaptation Solution

6.8.1 Power management adaptation

Power management in Debian is more complicated, and there are multiple standby ways in different systems, such as: the power_key.sh we added is button standby (pm-utils or writing nodes directly), pm-utils is command standby (writing nodes after executing the hook script).
In the power management of desktop systems such as xfce4, mate, and gnome, automatic standby, UI or button standby are generally given priority to systemd, and then consolekit or pm-utils systemd (writing nodes directly). Here, systemd uses pm utils firstly by default to realize the process of standby/wake up.

The `overlay/etc/Powermanager` in Debian has integrated related configuration files by default

```
├── 01npu
├── 02npu
├── 03wifibt
├── 04wifibt
├── power-key.conf
├── power-key.sh
├── triggerhappy
└── triggerhappy.service
```

There are some special handling of power management in the `etc/init.d/rockchip.sh`, especially the parts of NPU and RKWIFIBT, and the processing of power button is added at the same time.

```
# support power management
if [ -e "/usr/sbin/pm-suspend" -a -e /etc/Powermanager ] ;
then
    mv /etc/Powermanager/power-key.sh /usr/bin/
    mv /etc/Powermanager/power-key.conf /etc/triggerhappy/triggers.d/
    if [[ "$CHIPNAME" == "rk3399pro" ]];
```

```
    then
        mv /etc/Powermanager/01npu /usr/lib/pm-utils/sleep.d/
        mv /etc/Powermanager/02npu /lib/systemd/system-sleep/
    fi
    mv /etc/Powermanager/03wifibt /usr/lib/pm-utils/sleep.d/
    mv /etc/Powermanager/04wifibt /lib/systemd/system-sleep/
    mv /etc/Powermanager/triggerhappy /etc/init.d/triggerhappy

    rm /etc/Powermanager -rf
    service triggerhappy restart
fi
```

If there are other special modules that need wake-up processing, they can be placed in the following two configuration directories.

```
 /usr/lib/pm-utils/sleep.d/  and  /lib/systemd/system-sleep/
```

please refer to the `/rockchip-test/suspend_resume/suspend_resume.sh` for standby/wake-up test.

For other chip-related standby/wake-up processing, please refer to the development document `<SDK>/docs/cn/Common/TRUST/Rockchip_RK3588_Developer_Guide_System_Suspend_CN.pdf`.

6.8.2 Power Management Configuration

Here mainly introduces the power management configuration of xfce4, mainly related to DPMS and standby hibernation related configurations. The specific configuration of j is as follows:

```
#/etc/xdg/xfce4/xfconf/xfce-perchannel-xml/xfce4-power-manager.xml

<channel name="xfce4-power-manager" version="1.0">
  <property name="xfce4-power-manager" type="empty">
    <!-- Let other power manager(e.g. trigger-happy) handle power key -->
    <property name="power-button-action" type="uint" value="0"/>

    <!-- Disable DPMS(auto screen-off) by default -->
    <property name="dpms-enabled" type="bool" value="false"/>
    <property name="blank-on-ac" type="int" value="0"/>

    <!-- Disable Screen lock by default -->
    <property name="lock-screen-suspend-hibernate" type="bool" value="false"/>
  </property>
</channel>
```

6.9 Debian AI Adaptation Solution

NPU related adaptation on the Rockchip platform, RKNPU is mainly used to achieve model conversion, model inference, model performance evaluation functions, etc

6.9.1 RKNPU Adaptation

Mainly used for RK3568/RK3566/RK3588

```
rknpu2/
└── rknpu2.tar
```

6.9.2 RKNN Test Demo Adaptation

Mainly including NPU frequency conversion/pressure/demo related tests

```
/rockchip-test/npu2/
├── npu_freq_scaling.sh
├── npu_stress_test.sh
├── npu_test.sh
├── rknn_demo.sh
└── rknn_stress_test
```

6.10 Debian Firmware Upgrade Scheme

The main methods for Debian firmware upgrades are as follows:

- USB Upgrade
- SD Card or USB Drive Boot Upgrade
- Burner Upgrade
- OTA Upgrade

OTA upgrades are divided into Recovery Upgrade (including Loader, Parameter, other partitions, and customer-defined partitions), A/B Partition Upgrade, and Differential Upgrade.

For details, refer to `<SDK>/docs/cn/Linux/Recovery/Rockchip_Developer_Guide_Linux_Upgrade_CN.pdf`.

6.11 Debian Secure Boot Scheme

6.11.1 Secureboot

Secure boot is mainly divided into AVB and FIT methods, with the support status for each chip as follows:

Chip Name	Kernel Verification Method	Kernel Version	Storage Medium
RK3399/RK3288	AVB	4.4	eFuse
RK3308/RK3328/RK3326/PX30/RK3358	AVB	4.4	OTP
RK3588/RK3562/RK3566/RK3568	FIT	5.10	OTP

For details, refer to `<SDK>/docs/cn/Linux/Security/Rockchip_Developer_Guide_Linux_Secure_Boot_CN.pdf`.

6.12 Debian Touch Adaptation Solution

Xserver configuration needs to be configured, with maxrix related parameters configured

Or test it with the command line

xinput map-to-output

6.13 Debian USB Device Adaptation Solution

At present, usbdevice related services are added to support usb adb/acm/hid/mtp/ntb/rndis/uac1/uac2/ums/uvc and other functions

The system service is in `/lib/systemd/system/usbdevice.service`, and the main execution file is `/usr/bin/usbdevice`

You can configure usb-related functions through `/etc/profile.d/usbdevice.sh`, for example:

```
#!/bin/sh

# The env variables below can be overridden

# option: adb acm hid mtp ntb rndis uac1 uac2 ums uvc
export USB_FUNCS="adb"

export UMS_FILE=/userdata/ums_shared.img
export UMS_SIZE=256M
export UMS_FSTYPE=vfat
export UMS_MOUNT=0
export UMS_MOUNTPOINT=/mnt/ums
export UMS_RO=0
```

The debugging log is in `/tmp/usbdevice.log`, and the functions supported by USB can be checked through the configuration file `/tmp/.usbdevice`

6.14 Debian Sensor Adaptation Scheme

gsensor/lsensor...

- Install the lm-sensors package

```
  sudo apt-get install lm-sensors
```

- Detect sensors on the system

```
sudo sensors-detect
```

This will scan your system for sensors and prompt you to enable support for each type of sensor. Follow the prompts to enable support for the sensors you wish to use.

- Test the installation

```
sensors
```

6.15 Debian System Information

6.15.1 Debian Version Number

```
root@linaro-alip:~# cat /etc/debian_version
11.8
```

6.15.2 Debian Version Information

```
root@linaro-alip:~# cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 11 (bullseye)"
NAME="Debian GNU/Linux"
VERSION_ID="11"
VERSION="11 (bullseye)"
VERSION_CODENAME=bullseye
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
BUILD_INFO="root@pc 2022年 04月 27日 星期三 11:49:33 CST"
```

6.15.3 Debian System Hardware Information

```
root@linaro-alip:~# hardinfo  -r
Computer
 Summary
 Operating System
 Kernel Modules
 Boots
 Languages
 Filesystems
 Display
 Environment Variables
 Development
 Users
 Groups
...
```

6.15.4 Debian System Log Information

```
/info/
├── clk_summary -> /sys/kernel/debug/clk/clk_summary
├── cmdline -> /proc/cmdline
├── cpuinfo -> /proc/cpuinfo
├── device-tree -> /proc/device-tree
├── diskstats -> /proc/diskstats
├── dma_buf -> /sys/kernel/debug/dma_buf
├── dri -> /sys/kernel/debug/dri
├── fstab -> /etc/fstab
├── gpio -> /sys/kernel/debug/gpio
├── interrupts -> /proc/interrupts
├── iomem -> /proc/iomem
├── kallsyms -> /proc/kallsyms
├── log -> /var/log
├── meminfo -> /proc/meminfo
├── mountall.log -> /tmp/mountall.log
├── os-release -> /etc/os-release
├── partitions -> /proc/partitions
├── pinctrl -> /sys/kernel/debug/pinctrl/
├── rkcif-mipi-lvds -> /proc/rkcif-mipi-lvds
├── rk_dmabuf -> /proc/rk_dmabuf
├── rkisp0-vir0 -> /proc/rkisp0-vir0
├── slabinfo -> /proc/slabinfo
├── softirqs -> /proc/softirqs
├── version -> /proc/version
└── wakeup_sources -> /sys/kernel/debug/wakeup_sources

...
```

6.16 Debian Cropping

Debian requires more and more memory space as features are added in new release versions. Here are some component cropping for the desktop version on the Rockchip arm platform.

If you want to simplify the firmware, the following items can be cropped:

- base firmware package, such as

```
debian/ubuntu-build-service/bullseye-desktop-arm64/customization/package-
lists/linaro.list.chroot
```

- Newly added package for rockchip shell script

```
mk-rootfs-buster.sh
```

In addition,the following items can be cropped in the firmware:

- The libgl1-mesa-dri package of mesa, only kms_swrast_dri.so and swrast_dri.so are reserved under /usr/lib/*/dri/, others are useless.
- The linux-firmware package (if there is), /usr/lib/firmware, only keep the firmware you need (generally, customers do not need it)

- In packages, only keep the x11 mali package corresponding to your chip
- var/cache

In addition, you can also adjust the image file size of dd in mk-image.sh when making an image, for example

```
truncate -s 2500M rootfs.img
mkfs.ext4 -d binary rootfs.img
```

The general space occupied by the system is as follows:

```
root@pc:/# du -sh */
120M    bin/
22M boot/
12K dev/
10M etc/
20K home/
1.6G    lib/
4.0K    media/
4.0K    mnt/
4.0K    opt/
4.0K    proc/
36M rockchip-test/
28K root/
44K run/
26M sbin/
4.0K    srv/
4.0K    sys/
18M system/
4.0K    tmp/
1.3G    usr/
73M var/
8.0K    vendor/
```

6.17 Debian Testing

- Integrate Rockchip stress test script

The rockchip_test integrates functional, stress, and performance related tests:

```
root@linaro-alip:~# /rockchip-test/rockchip_test.sh

cpu test:             1 (cpufreq stresstest)
ddr test:             2 (memtester & stressapptest)
gpu test:             3 (use glmark2)
npu test:             4 (npu2:rk3588)
auto reboot test:     5 (reboot tests)
suspend_resume test:  6 (suspend & resume)
nand power lost test: 7 (S5 stress tests)
flash stress test:    8 (flash tests)
audio test:           9 (audio tests)
recovery test:        10 (default wipe all)
bluetooth test:       11 (bluetooth on&off test)
wifi test:            12 (wifi on&off test)
ethernet test:        13 (ethernet tests)
```

```
camera test:          14 (use rkaiq_demo)
video test:           15 (use gstreamer-wayland and app_demo)
chromium test:        16 (chromium with video hardware acceleration)
hardware infomation:  17 (to get the hardware infomation)

**************************************************
please input your test moudle:
```

6.18 Debian Debugging Tools

6.18.1 ADB Tool

6.18.1.1 ADB Overview

- Running a shell on the device (command line)
- Managing port mapping for emulators or devices
- Uploading/downloading files between a computer and a device
- Installing local software onto Debian devices
- ADB is a "client-server" program, where the client mainly refers to the PC, and the server side is the physical machine or virtual machine of the Debian device. Depending on how the PC is connected to the Debian device, ADB can be divided into two types:
- Network ADB: The host connects to the STB device via wired/wireless network (in the same LAN)
- USB ADB: The host connects to the STB device via a USB cable

6.18.1.2 USB ADB Usage Instructions

USB ADB has the following limitations:

- Only supports USB OTG port
- Does not support multiple clients using it simultaneously
- Only supports the host connecting to one device, does not support connecting to multiple devices
  The connection steps are as follows:
  To test if the connection is successful, run the "adb devices" command. If the device's serial number is displayed, it indicates a successful connection.

6.18.2 systemd-analyze for Analyzing Boot Performance

`systemd-analyze` is a tool for analyzing boot performance, used to analyze the time consumption of services during startup

- Displays the time consumed by each initialization step

```
root@linaro-alip:/# systemd-analyze time
Startup finished in 1.523s (kernel) + 3.833s (userspace) = 5.357s
graphical.target reached after 3.810s in userspace
```

- Lists the initialization time of all units

```
root@linaro-alip:/# systemd-analyze blame
```

```
1.680s rkwifibt.service
1.368s keyboard-setup.service
1.148s console-setup.service
 588ms blueman-mechanism.service
 557ms dev-mmcblk0p6.device
 495ms usbdevice.service
 319ms udisks2.service
 287ms resize-all.service
 283ms systemd-udev-trigger.service
 246ms systemd-journal-flush.service
 203ms user@0.service
 198ms user@1000.service
 196ms NetworkManager.service
 185ms nginx.service
 185ms systemd-rfkill.service
 184ms systemd-journald.service
 ...
```

6.18.3 systemd-analyze analyzes boot performance

`systemd-analyze` is a tool for analyzing boot performance, used to analyze the time consumption of services during startup.

- Displays the time consumed by each initialization step

```
root@linaro-alip:/# systemd-analyze time
Startup finished in 1.523s (kernel) + 3.833s (userspace) = 5.357s
graphical.target reached after 3.810s in userspace
```

- Lists the initialization time of all units

```
root@linaro-alip:/# systemd-analyze blame

1.680s rkwifibt.service
1.368s keyboard-setup.service
1.148s console-setup.service
 588ms blueman-mechanism.service
 557ms dev-mmcblk0p6.device
 495ms usbdevice.service
 319ms udisks2.service
 287ms resize-all.service
 283ms systemd-udev-trigger.service
 246ms systemd-journal-flush.service
 203ms user@0.service
 198ms user@1000.service
 196ms NetworkManager.service
 185ms nginx.service
 185ms systemd-rfkill.service
 184ms systemd-journald.service
 ...
```

6.18.4 perf Performance Tuning Tool

The perf performance tuning tool, introduced as follows:

```
apt install -fy linux-perf-5.10

root@linaro-alip:/# perf

 usage: perf [--version] [--help] [OPTIONS] COMMAND [ARGS]

 The most commonly used perf commands are:
   annotate        Read perf.data (created by perf record) and display annotated
code
   archive         Create archive with object files with build-ids found in
perf.data file
   bench           General framework for benchmark suites
   buildid-cache   Manage build-id cache.
   buildid-list    List the buildids in a perf.data file
   c2c             Shared Data C2C/HITM Analyzer.
   config          Get and set variables in a configuration file.
   data            Data file related processing
   diff            Read perf.data files and display the differential profile
   evlist          List the event names in a perf.data file
   ftrace          simple wrapper for kernel's ftrace functionality
   inject          Filter to augment the events stream with additional
information
   kallsyms        Searches running kernel for symbols
   kmem            Tool to trace/measure kernel memory properties
   kvm             Tool to trace/measure kvm guest os
   list            List all symbolic event types
   lock            Analyze lock events
   mem             Profile memory accesses
   record          Run a command and record its profile into perf.data
   report          Read perf.data (created by perf record) and display the
profile
   sched           Tool to trace/measure scheduler properties (latencies)
   script          Read perf.data (created by perf record) and display trace
output
   stat            Run a command and gather performance counter statistics
   test            Runs sanity tests.
   timechart       Tool to visualize total system behavior during a workload
   top             System profiling tool.
   version         display the version of perf binary
   probe           Define new dynamic tracepoints
   trace           strace inspired tool
```

7. Debian Security Updates

Debian security updates are as follows:

Debian-security

Additionally, you can conveniently receive the latest security updates using APT. To keep your Debian operating system up to date with the latest security updates, add the following line to your `/etc/apt/sources.list` file:

```
deb http://security.debian.org/debian-security bookworm-security main contrib
non-free non-free-firmware
```

After saving the changes, execute the following two commands to download and install security updates:

```
apt-get update && apt-get upgrade
```

If you discover a security issue in your packages or in someone else's, be sure to contact the security team via the email address `team@security.debian.org`. They track existing security issues, can assist maintainers in resolving security problems or fix the issues themselves, and are responsible for sending security advisories and maintaining security.debian.org.

Latest Alerts
The following page is a brief archive of security alerts published on the debian-security-announce mailing list.

```
[2023/08/07] DSA-5471-1 libhtmlcleaner-java security update
[2023/08/06] DSA-5470-1 python-werkzeug security update
[2023/08/05] DSA-5469-1 thunderbird security update
[2023/08/05] DSA-5468-1 webkit2gtk security update
[2023/08/04] DSA-5467-1 chromium security update
[2023/08/04] DSA-5466-1 ntpsec security update
[2023/08/03] DSA-5465-1 python-django security update
[2023/08/03] DSA-5464-1 firefox-esr security update
[2023/07/30] DSA-5463-1 thunderbird security update
[2023/07/30] DSA-5462-1 linux security update
[2023/07/30] DSA-5461-1 linux security update
[2023/07/26] DSA-5460-1 curl security update
[2023/07/25] DSA-5459-1 amd64-microcode security update
[2023/07/25] DSA-5458-1 openjdk-17 security update
[2023/07/22] DSA-5457-1 webkit2gtk security update
[2023/07/20] DSA-5456-1 chromium security update
[2023/07/17] DSA-5455-1 iperf3 security update
[2023/07/16] DSA-5454-1 kanboard security update
[2023/07/16] DSA-5453-1 linux security update
[2023/07/14] DSA-5452-1 gpac security update
[2023/07/09] DSA-5451-1 thunderbird security update
```

8. Debian Roadmap

Currently, the official RK support for Debian versions ranges from stretch (9) to bookworm (12), with the primary maintenance focus on bullseye (11).

Version	Debian Released	SDK Released	EOL LTS
Stretch (9)	2017-06-17	2018-06-20	2022-07-01
Buster (10)	2019-07-06	2020-12-11	2024-06-30
Bullseye (11)	2021-08-14	2022-01-15	~ 2026
Bookworm (12)	2023-06-10	2023 Q4	~ 2028

Debian LTS versions are updated every two years.

For more details, refer to the official Debian release information:

Debian-Release

9. Debian FAQ

This chapter is going to answer some frequently asked questions about Debian GNU/Linux based on the Rockchip platform. For other questions, please refer to the official website Debian FAQ.

9.1 "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 path of project's directory, 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.

9.2 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:

To uses live build in Debian, configured like followings to change the image source to domestic:

```
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 .

9.3 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 way:

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

9.4 Multiple Mounts lead to /dev error

For example: sudo: unable to allocate pty: No such device appears

The reason may be that the compilation process has been mounted multiple times, resulting in the above error, which can be fixed by the following way:

```
ssh <username>@<IP address> -T sudo -S umount /dev -l
```

9.5 How to Check System Related Information

9.5.1 How to Check the Debian Version of Your System

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

9.5.2 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
```

9.5.3 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
```

9.5.4 The ssh.service Is Abnormal in System

This is a problem of Debian10 or earlier version, please add the following code in the /etc/rc.local:

```
#!/bin/sh -e
#
# rc.local
#
```

```
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
# Generate the SSH keys if non-existent
if [ ! -f /etc/ssh/ssh_host_rsa_key ]
then
    # else ssh service start in dpkg-reconfigure will fail
    systemctl stop ssh.socket||true
    dpkg-reconfigure openssh-server
fi

exit 0
```

9.6 Debian11 Base Package Fails to Build

An error similar to the following will be encountered:

```
W: Failure trying to run: /sbin/ldconfig
W: See //debootstrap/debootstrap.log for details
```

It is mainly required that the kernel version of the PC should be 5.10+, which is a bug in the old QEMU. There are two solutions:

- The kernel version that comes with the PC must meet the requirements of 5.10+.

The way to check PC's Kernel Version:

```
cat /proc/version
Linux version 5.13.0-39-generic
```

- Update system's qemu

Please refer to qemu.

9.7 How to Decompress, Modify and Repackage Debian deb Package

If you want to modify and repackage on the original deb, please refer to the follow way:

```
#Decompress the files in the package to the extract directory
dpkg -X xxx.deb extract/

#Decompress the control information of the package under extract/DEBIAN/:
dpkg -e xxx.deb extract /DEBIAN/

#Modify the file XXX

# Repackage the modified content to generate a deb package
dpkg-deb -b extract/ .
```

9.8 How to Add the Swap Partition in Debian

When the physical memory of the system is not enough, you can add Debian's swap virtual memory partition for the current running program. For example, create a 2G virtual memory

- Create a swap file

```
cd /opt
mkdir swap
dd if=/dev/zero of=swapfile bs=1024 count=2000000
# count represents the size, here is 2G.
```

- Convert files to swap files

```
sudo mkswap swapfile
```

- Activate the swap file

```
swapon /opt/swapfile

Uninstall:
swapoff /opt/swapfile
```

If it is automatically mounted after booting, you can add it to the /etc/fstab file
eg : /opt/swapfile swap swap defaults 0 0

- Verify whether it is in effect

```
root@linaro-alip:/opt# free -h
               total        used        free      shared  buff/cache   available
memory:      1.9Gi       390Mi        91Mi        75Mi       1.5Gi       1.4Gi
swap:      1.9Gi          0B       1.9Gi
e =h
```

9.9 Update Debian System for the First Time Will Restart the Display Service

In general, in order to be compatible with different chips, when Debian starts for the first time, /etc/init.d/rockchip.sh will install various differential packages according to the chip, such as libmali isp and other packages. After installation, the display service will be restarted. If it is an independent project, it can be placed in the image to process this difference.

9.10 Errors Occurring in Debian When Calling libGL related dri.so

Introduction as follows:

- EGL is an extension of OpenGL on the ARM platform for the x window system, and its function is equivalent to the glx library under x86.
- Since the driver modesettings used by Xorg will load libglx.so by default (disabling glx will cause some applications which detecting through glx environment fail to start), libglx.so will search for the dri library in the system. However, Xorg 2D acceleration is implemented directly based on DRM and does not implement the dri library, so libglx.so will report the following error during booting.

```
AIGLX error: dlopen of /usr/lib/aarch64-linux-gnu/dri/rockchip_dri.so failed`
```

It has no influence on system operation, please ignore it.

Similarly, the following errors will also be reported during the booting process of some applications, please ignore it for it has not influence on applications operation.

```
libGL error: unable to load driver: rockchip_dri.so
libGL error: driver pointer missing
libGL error: failed to load driver: rockchip
```

9.11 How to Confirm that the Hardware Mouse Layer is Useful in Debian

- Configure kernel dts

Similar to the following log:

```
root@linaro-alip:~# dmesg |grep cursor
[ 2.062561] rockchip-vop2 fe040000.vop: [drm:vop2_bind] Cluster1-win0 as cursor
plane for vp0
[ 2.062669] rockchip-vop2 fe040000.vop: [drm:vop2_bind] Cluster0-win0 as cursor
plane for vp1
```

- modetest test whether the layer has been reported

```
102     0       0       0,0             0,0     0               0x00000002
  formats: XR30 AR30 XB30 AB30 XR24 AR24 XB24 AB24 RG24 BG24 RG16 BG16 YU08 YU10
YUYV Y210
  props:
        8 type:
                flags: immutable enum
                enums: Overlay=0 Primary=1 Cursor=2
                value: 2
```

- Check if the summary has called the hardware mouse layer

root@linaro-alip:~# cat /sys/kernel/debug/dri/0/summary |grep 64x64

If there are steps 1/2, and there are still problems, then check whether /var/log/drm-cursor.log has abnormalities.

9.12 The log is Too Large in Debian

Debian provides **logrotate** to manage log files. Logrotate is intended to simplify log file management for systems that will generate many log files. Logrotate supports automatic rotation compression, deletion and sending log related emails. Logrotate can be run daily, weekly, monthly or when the log file size reaches a certain value. Typically, logrotate is run as a daily cron job.

```
apt install -fy logrotate cron
```

9.13 Debian Setting Multi User Mode Issue

When the system starts, the 'systemd' process will try to start '/lib/systemd/system/default. target' (usually the graphical interface system is a symbolic link to "graphical. target"). The status can be obtained through the following command:

```
systemctl get-default graphical.target
```

Setting multi user mode (command line system):

```
systemctl set-default multi-user.target
```

After restarting, it was found that the interface was suspended at the logo and could not access the system.
The normal startup sequence of the system is' sysinit ->multi user ->graphic '. If it is set to' multi user. target ', it means that the graphical interface has not been started. In this case, VT2 (terminal interaction needs to be enabled) is required, which means that the kernel needs to open the following two macros

```
CONFIG_ FRAMEBUFFER_ CONSOLE=y
CONFIG_ VT=y
```

9.14 Debian Username and Password

The default username and password for the system are 'linaro' and 'linaro',
`Root 'can log in without a password, and can be accessed through the command line to' sudo su '`

9.15 Debian XFCE Desktop icon Double-click Abnormal

XFCE desktop native bug can be resolved by checking 'Settings ->Desktop ->Icon' and clicking on 'Activate Project'.

9.16 Chromium browsers will have a command line flag: --no-sandbox

Unless using non-hardware accelerated, official native browser version. Otherwise, the customized browser version needs to be started with the `-no-sandbox` parameter, because the sandbox is a permission management and controls file access, and only without a sandbox can access to the hardware node is allowed to achieve hardware acceleration.

9.17 Setting Up DRI2 Extension in Debian System's X11

In the SDK, `glmark2-es2` uses the `dri2` interface for display with the Mali GPU library.

For specific implementation, refer to the relevant code in `xserver`:

- File Path: `./hw/xfree86/drivers/modesetting/dri2.c`
- Functions: `ms_dri2_get_msc` and `ms_dri2_schedule_wait_msc`

To confirm if Xserver supports DRI2, check the following log:

```
root@linaro-alip:/# grep -i dri2 /var/log/Xorg.0.log
[    47.696] (II) modeset(0): [DRI2] Setup complete
[    47.699] (II) modeset(0): [DRI2]   DRI driver: rockchip
[    47.712] (II) modeset(0): [DRI2]   VDPAU driver: rockchip
[    48.502] (II) Initializing extension DRI2
```

Below is a segment of the test code named `dri2-test.c` for DRI2.

```
#include <stdlib.h>
#include <stdio.h>
#include <xcb/xcb.h>
#include <xcb/dri2.h>
#include <X11/Xlib.h>
#include <X11/Xlib-xcb.h>

int main(void)
{
    xcb_connection_t *c;
    xcb_dri2_connect_cookie_t cookie;
    xcb_dri2_connect_reply_t *reply;
    Display *display = XOpenDisplay(NULL);
```

```
    Window window = DefaultRootWindow(display);
    c = XGetXCBConnection(display);
    cookie = xcb_dri2_connect(c, window, XCB_DRI2_DRIVER_TYPE_DRI);
    reply = xcb_dri2_connect_reply(c, cookie, 0);
    printf("%s[%d] device(%s)\n", __func__, __LINE__,
xcb_dri2_connect_device_name (reply));
    c = xcb_connect(NULL, NULL);
    xcb_screen_t *screen = xcb_setup_roots_iterator(xcb_get_setup(c)).data;
    cookie = xcb_dri2_connect(c, screen->root, XCB_DRI2_DRIVER_TYPE_DRI);
    reply = xcb_dri2_connect_reply(c, cookie, 0);
    printf("%s[%d] device(%s)\n", __func__, __LINE__,
xcb_dri2_connect_device_name (reply));
    return 0;
}

build and test:

# gcc dri2-test.c -lxcb -lxcb-dri2 -lX11 -lX11-xcb -o dri2-test
# ./dri2-test
main[21] device(/dev/dri/card0)
main[27] device(/dev/dri/card0)
```

9.18 Installing GCC Toolchain on Debian

To install, run the following command:

```
apt update && apt install -y build-essential manpages-dev
```

After installation, confirm the GCC version:

```
root@linaro-alip:/# gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/aarch64-linux-gnu/10/lto-wrapper
Target: aarch64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Debian 10.2.1-6' --with-
bugurl=file:///usr/share/doc/gcc-10/README.Bugs --enable-
languages=c,ada,c++,go,d,fortranx
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 10.2.1 20210110 (Debian 10.2.1-6)
```

9.19 Auto-Completion for Installing Packages on Debian

This is a common operation in Linux systems. Normally, you need to install the `bash-completion` package.

For specific instructions, search online, but it generally involves:

```
sudo apt-get install bash-completion
source /etc/bash_completion
```

9.20 Supporting DRI3 Extension in Debian X11 System

DRI3 extension, part of X11, provides improved support for direct rendering. It's usually enabled by default in Debian. As a low-level protocol, it's not directly executed without external calls.

To use the DRI3 extension in applications, refer to DRI3 documentation and write code to call the respective interfaces. For example, use DRI3 interfaces provided by the XCB (X protocol C-language Binding) library. Here's how to install and view these interfaces in Debian:

- **Installing XCB DRI3 Development Libraries**

  Install the `libxcb-dri3-dev` library with:

  ```
  sudo apt-get install libxcb-dri3-dev
  ```

  This installs the necessary libraries and headers for using DRI3 functions in your applications.

- **Viewing DRI3 Interfaces**

  To view the definitions and functions of DRI3 interfaces, check the `dri3.h` header file:

  ```
  vi /usr/include/xcb/dri3.h
  ```

  This file contains specific implementations of the DRI3 interface, helping you understand how to use the DRI3 extension in your applications.

Follow these steps to start using the DRI3 extension in Debian for developing or improving your graphics applications. Remember to refer to official XCB and DRI3 documentation for correct and efficient use of these interfaces.

9.21 Setting System to Boot into Command Line Mode

The general method involves:

- Uninstalling xserver
- Enabling `CONFIG_FRAMEBUFFER_CONSOLE`, `CONFIG_DRM_FBDEV_EMULATION`, `CONFIG_VT`, `CONFIG_VT_CONSOLE` in the kernel

9.22 Configuring Screen Rotation

Refer to `/etc/X11/xorg.conf.d/20-modesetting.conf`

```
...
Section "Screen"
    Identifier  "Default Screen"
    Device      "Rockchip Graphics"
    Monitor     "Default Monitor"
    DefaultDepth        24
    SubSection "Display"
            Depth       24
            Modes       "1024x600"
    EndSubSection
EndSection
```

```
### Valid values for rotation are "normal", "left", "right"
Section "Monitor"
    Identifier  "HDMI-A-1"
    Option      "Rotate" "inverted"
    Option      "Position" "0x0"
EndSection
Section "Monitor"
    Identifier  "DSI-1"
    Option      "Rotate" "left"
    Option      "Position" "0x0"
EndSection
```

9.23 Implementing No Black Screen Function in Debian

Black screen refers to the time from X service startup to desktop application display (dependent on the desktop application itself).

To maintain a logo during this time, add the following before executing `Xorg.wrap` in `/usr/bin/X`:

```
export XSERVER_FREEZE_DISPLAY=/.freeze_xserver
touch $XSERVER_FREEZE_DISPLAY
$(sleep 6; rm $XSERVER_FREEZE_DISPLAY)&
```

Freeze for 6 seconds, then display the desktop. Adjust the freeze duration as needed.

Alternatively:

```
{
        export XSERVER_FREEZE_DISPLAY=/.freeze_xserver
        touch $XSERVER_FREEZE_DISPLAY
        while sleep .5; do
                pgrep panel && break # Waiting for status bar service
        done
        sleep 2 # Waiting for status bar rendering
        rm $XSERVER_FREEZE_DISPLAY
}&
```

9.24 Removing Desktop Mouse Pointer Display in Debian System

The native mechanism is designed for compatibility with some non-touch applications, where Xserver converts the first touch event into a mouse event.

To bypass, try:

- Using a transparent mouse theme (search online for specifics)
- Modify `hw/xfree86/drivers/modesetting/drmmode_display.c` source code, removing `drmModeSetCursor`, `drmModeMoveCursor` calls
- If the SDK includes a drm-cursor library, modify `/etc/drm-cursor.conf` and add `hide=1`

9.25 Steps for Compiling and Porting the rkaiq/rkisp Repository in Debian

To port the `rkaiq/rkisp` functionality from Buildroot to Debian:

9.25.1 Overview of Steps

For chips like RK3588, switch to a lower version of GCC and GLIBC for porting to a third-party system.

Step 1: Modify the Buildroot configuration to support GCC 8.

```
[Various changes to the configuration files, including switching from
BR2_cortex_a76_a55 to BR2_cortex_a72_a53 and setting the GCC version to 8.x]
```

Step 2: Set the Buildroot configuration to use GCC 8 and GLIBC 2.28 by default.

```
buildroot# cat configs/rockchip_rk3588_glibc2.28_defconfig

[Configuration for rockchip_rk3588 with GCC version 8.x and GLIBC 2.28]
```

Step 3: Compile with Buildroot using the configured settings.

Set up the Buildroot environment and compile the `camera-engine-rkaiq` module.

```
<SDK># source buildroot/envsetup.sh rockchip_rk3588_glibc2.28
<SDK>## cd buildroot
buildroot# make camera-engine-rkaiq
```

After compilation, port the generated files (e.g., `output/rockchip_rk3588_glibc2.28/build/camera-engine-rkaiq-1.0/camera-engine-rkaiq-1.0.tar`) to the Debian system.

9.26 How to Download Offline Deb Packages in Debian

```
root@linaro-alip:/# apt-get download <package name>
```

9.27 How to Check glibc Version in Debian

```
root@linaro-alip:/# ldd --version
ldd (Debian GLIBC 2.31-13+deb11u7) 2.31
or
# /lib/libc.so.6
GNU C Library (GNU libc) stable release version 2.35.
```

9.28 Support for Screen Splitting in Debian Systems

For a physically split screen with a resolution of 3840x2160 cut into 3840x720, the output resolution must be 3840x2160 to light up the screen, but only the 720 part is displayed.

The latest xserver added padding support.

Setting method:

```
+++ b/overlay/etc/X11/xorg.conf.d/20-modesetting.conf
[Configuration changes to set VirtualSize and Padding for the display]
```

Modify `/etc/X11/xorg.conf.d/20-modesetting.conf` as above. `VirtualSize` is configured as the desired resolution, and `Padding` is configured for the pixels cut off in the order top, bottom, left, right.

For unclear text on the screen, modify the padding order.

```
Option "VirtualSize" "DSI-1:1920x316"
Option "Padding" "DSI-1:0,0,0,764"
```

Mouse movement is relative and usually doesn't require special modifications. If there are ratio or position issues, consider using a software cursor (remove the mouse layer in kernel dts configuration or configure SWcursor in the modesetting conf).

To remove the cursor in kernel dts, search and delete, like:

```
kernel/arch/arm64/boot/dts/rockchip# ag cursor
rk3566-evb2-lp4x-v10-linux.dts
12:     cursor-win-id = <ROCKCHIP_VOP2_CLUSTER0>;
```

Or configure SWcursor in modesetting conf:

```
--- a/overlay/etc/X11/xorg.conf.d/20-modesetting.conf
+++ b/overlay/etc/X11/xorg.conf.d/20-modesetting.conf
@@ -2,6 +2,8 @@ Section "Device"
     Identifier  "Rockchip Graphics"
     Driver      "modesetting"

+    Option      "SWcursor"       "TRUE"
+
```

10. Debian Third-Party Open Source Software and License Information

For information on open source licensing related to Debian, refer to the official website's legal section.

11. Debian Reference Materials

For official Debian documentation, consult the Debian Documentation.

For information on official stable packages in Debian, visit the Debian Packages section.