Rockchip Buildroot Weston Developer Guide

ID: RK-KF-YF-326

Release Version: V1.4.0

Release Date: 2022-05-09

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

DISCLAIMER

THIS DOCUMENT IS PROVIDED "AS IS". ROCKCHIP ELECTRONICS CO., LTD.("ROCKCHIP")DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS,MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

Trademark Statement

"Rockchip", "瑞芯微", "瑞芯" shall be Rockchip's registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

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 presents the basic configuration methods of Buildroot SDK Weston display service.

Product Version

Chipset	Kernel Version
All chipset	All

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers
Software development engineers

Revision History

Version	Author	Date	Revision History
V1.0.0	Jeffy Chen	2019-11-27	Initial version
V1.0.1	Ruby Zhang	2020-07-23	Update the company name and the format of the document
V1.1.0	Jeffy Chen	2020-08-04	Update contents for the newest SDK
V1.2.0	Jeffy Chen	2020-09-25	Update contents for the newest SDK
V1.3.0	Jeffy Chen	2022-01-13	Update contents for the newest SDK
V1.4.0	Jeffy Chen	2022-05-09	Update contents for the newest SDK

Contents

1. Introduction

1.1 Overview

Weston is the official implementation reference of Wayland open source display protocol, and Weston 10.0.0 drm back-end is used in Rockchip Buildroot SDK by default.

1.2 Configuration Methods

There are multiple ways to configure Rockchip Buildroot SDK Weston:

a. Command line options

That is, the options of the command when starting Weston, such as weston --tty=2.
It's located in /etc/init.d/S49weston, corresponding to the location of SDK code:
buildroot/package/weston/S49weston.

b. weston.ini configuration file

It's located in /etc/xdg/weston/weston.ini and /etc/xdg/weston/weston.ini.d/, corresponding to the location of SDK code: buildroot/board/rockchip/common/base/etc/xdg/weston/weston.ini.

Please refer to: https://fossies.org/linux/weston/man/weston.ini.man.

c. Special environment variables

Generally, these environment variables are set in /etc/profile.d/weston.sh, corresponding to the location of SDK code: buildroot/package/weston/weston.sh.

d. Dynamic configuration file

For drm back-end, Buildroot SDK Weston provides some dynamic configuration support, the default path is /tmp/.weston_drm.conf, which can be specified by the environment variable WESTON_DRM_CONFIG.

e. udev rules

Part of the configurations of input devices in Weston should be set by `udev rules`.

2. Configuration

2.1 Identify screens

The Weston is using output(head)'s name to identify screens. The names can be found through:

1. Weston's startup log:

```
# weston&
[02:11:29.746] DRM: head 'DSI-1' found ...
```

2. weston-info/wayland-info tools:

```
# weston-info
...
output: 15
name: 'DSI-1'
...
```

2.2 Cursor Configuration

Weston supports setting the theme and size of cursor in the `shell` section of weston.ini configuration file, for example:

```
# /etc/xdg/weston/weston.ini

[shell]
cursor-theme=whiteglass # Buildroot SDK supports
comix/obsidian/xcursor/xcursor-transparent cursor theme packages.
cursor-size=24
```

2.3 Status Bar Configuration

Weston supports setting the background color and position of status bar in the `shell` section of weston.ini configuration file, and setting the quick start program in the `launcher` section, for example:

```
# /etc/xdg/weston/weston.ini

[shell]
panel-color=0x90ff0000
# the color format is ARGB8888

panel-position=bottom
# top|bottom|left|right|none,none is disable

[launcher]
icon=/usr/share/icons/gnome/24x24/apps/utilities-terminal.png
# icon path

path=/usr/bin/gnome-terminal
# quick start command
```

Currently, Weston does not support setting the size of status bar. You have to modify in the code level when need some adjustments:

```
// weston/clients/desktop-shell.c
```

```
    static void
    panel_configure(void *data,
                    struct weston_desktop_shell *desktop_shell,
                    uint32_t edges, struct window *window,
                    int32_t width, int32_t height)
    {
            ...
            switch (desktop->panel_position) {
            case WESTON_DESKTOP_SHELL_PANEL_POSITION_TOP:
            case WESTON_DESKTOP_SHELL_PANEL_POSITION_BOTTOM:
                    height = 32; # height
                    break;
            case WESTON_DESKTOP_SHELL_PANEL_POSITION_LEFT:
            case WESTON_DESKTOP_SHELL_PANEL_POSITION_RIGHT:
                    switch (desktop->clock_format) {
                    case CLOCK_FORMAT_NONE:
                            width = 32;
                            break;
                    case CLOCK_FORMAT_MINUTES:
                            width = 150;
                            break;
                    case CLOCK_FORMAT_SECONDS:
                            width = 170;
                            break;
                    }
                    break;
            }
```

2.4 Background Configuration

Weston supports setting the background pattern and color in the `shell` section of the weston.ini configuration file, such as

```
# /etc/xdg/weston/weston.ini

[shell]
background-image=/usr/share/backgrounds/gnome/Aqua.jpg
# Background pattern (wallpaper) absolute path

background-type=tile
# scale|scale-crop|tile

background-color=0xff002244
# The color format is ARGB8888, effective when no background pattern is set
```

2.5 Idle Time and Lock Screen Configuration

The idle timeout of Weston can be configured in the command options or in the `core` section of weston.ini, such as:

```
# /etc/init.d/S49weston
start_weston()
{
  /usr/bin/weston --idle-time=0&  # 0 means idle mode is disabled, in seconds
}
```

Or

```
# /etc/xdg/weston/weston.ini


[core]
idle-time=10
```

Lock screen of Weston can be configured in the `shell` section of weston.ini, such as:

```
# /etc/xdg/weston/weston.ini


[shell]
locking=false
# lock screen is disabled

lockscreen-icon=/usr/share/icons/gnome/256x256/actions/lock.png
# unlock button icon

lockscreen=/usr/share/backgrounds/gnome/Garden.jpg
# background of lock screen
```

2.6 Color Format Configuration

The default display format of Weston in the Buildroot SDK is ARGB8888. For some low-performance platforms, you can configure RGB565 in the `core` section in the weston.ini, such as:

```
# /etc/xdg/weston/weston.ini


[core]
gbm-format=rgb565
# xrgb8888|rgb565|xrgb2101010
```

You can also configure the display format of each screen individually in the `output` section of weston.ini, such as:

```
# /etc/xdg/weston/weston.ini


[output]
name=LVDS-1

gbm-format=rgb565
# xrgb8888|rgb565|xrgb2101010
```

2.7 Display Orientation Configuration

Display orientation of screens can be configured in the `output` section of weston.ini, such as

```
# /etc/xdg/weston/weston.ini

[output]
name=LVDS-1

transform=90
# normal|90|180|270|flipped|flipped-90|flipped-180|flipped-270
```

If you want to configure the orientation dynamically, the dynamic configuration file can be used, such as:

```
echo "output:all:rotate90" > /tmp/.weston_drm.conf # All screens rotate 90
degrees
echo "output:eDP-1::rotate180" > /tmp/.weston_drm.conf # eDP-1 rotates 180
degrees
```

2.8 Resolution and Scale Configuration

Screen's resolution and scale of Weston can be configured in the `output` section of weston.ini, such as:

```
# /etc/xdg/weston/weston.ini

[output]
name=LVDS-1

mode=1280x800
# should be an effective resolution supported by the screen

scale=2
# value must be an integer, support application level scaling
```

If you want to scale to a specific resolution(without changing the physical resolution), you can configure it for all screens through WESTON_DRM_VIRTUAL_SIZE environment variable, such as:

```
# /etc/profile.d/weston.sh
export WESTON_DRM_VIRTUAL_SIZE=1024x768
```

If you want to configure resolution and scaling dynamically, the dynamic configuration file can be used, for example:

```
echo "output:HDMI-A-1:mode=800x600" > /tmp/.weston_drm.conf # change the
resolution of HDMI-A-1 to 800x600
echo "output:eDP-1:rect=<10,20,410,620>" > /tmp/.weston_drm.conf # eDP-1
display to the position of (10,20), the size is scaled to 400x600
echo "output:HDMI-A-1:size=1920x1080" > /tmp/.weston_drm.conf # HDMI-A-1
scale to 1080p, the physical resolution is unchanged.
```

When the VOP hardware doesn't support scaling, these kinds of scale requires Rockchip's RGA 2D acceleration.

2.9 Freeze the Screen

When Weston is started, there will be a black screen that switches between boot logo and UI display temporarily. If you want to prevent this black screen, you can freeze the Weston screen content temporarily through the following ways:

Add --warm-up to Weston's command options

```
# /etc/init.d/S49weston
start_weston()
{
  /usr/bin/weston --warm-up&
}
```

Or

```
# /etc/init.d/S49weston
start_weston()
{
  export WESTON_FREEZE_DISPLAY=/tmp/.weston_freeze # Set a special
configuration file path
  touch /tmp/.weston_freeze # Freeze the display
  /usr/bin/weston&
  sleep 1 && rm /tmp/.weston_freeze& # unfreeze after 1 second
}
```

Or

```
# /etc/init.d/S49weston
start_weston()
{
  echo "output:all:freeze" > /tmp/.weston_drm.conf # Freeze the display
  /usr/bin/weston&
  ...
  sleep 1 && \
  echo "output:all:unfreeze" > /tmp/.weston_drm.conf& # unfreeze  after 1
second
}
```

2.10 Screen Status Configuration

The DRM framework supports setting force status for screens:

```
echo on > /sys/class/drm/card0-HDMI-A-1/status # Force HDMI-A-1 connected
#on|off|detect, detect means hot-plug
```

For more specific screen status configuration, the dynamic configuration file can be used, for example:

```
    echo "output:DSI-1:off" > /tmp/.weston_drm.conf # Turn off DSI
    echo "output:HDMI-A-1:freeze" > /tmp/.weston_drm.conf # Freeze HDMI-A-1
    echo "output:eDP-1:on" > /tmp/.weston_drm.conf # Turn on eDP
    echo "compositor:state:off" > /tmp/.weston_drm.conf # Turn off display
    echo "compositor:state:sleep" > /tmp/.weston_drm.conf # Turn off display and
touch to wake
    echo "compositor:state:on" > /tmp/.weston_drm.conf # Turn on display
```

2.11 Multi-screen Configuration

The Buildroot SDK Weston supports mirror displays, extend displays, screen position configuration and hot plug functions.

For mirrored display, it would try to use Rockchip's RGA 2D acceleration when the VOP hardware doesn't support scaling.

Those configurations can be set by environment variables, for example:

```
    # /etc/profile.d/weston.sh
    export WESTON_DRM_PRIMARY=HDMI-A-1 # Specify HDMI-A-1 as primary monitor
    export WESTON_DRM_SINGLE_HEAD=1 # Force using single monitor
    export WESTON_DRM_MIRROR=1 # In mirror mode (multi-screen with the same
 display), without setting this environment variable will be with different
 display
    export WESTON_DRM_KEEP_RATIO=1 # In mirror mode, scaling maintains the aspect
 ratio, without setting this variable will be full screen by force
    export WESTON_DRM_HEAD_MODE=primary # Only enable primary monitor
    export WESTON_DRM_HEAD_MODE=internal # Only enable internal monitors
    export WESTON_DRM_HEAD_MODE=external # Only enable external monitors
    export WESTON_DRM_HEAD_MODE=external-dual # Enable all monitors, and prefer
 the external ones
    export WESTON_DRM_HEAD_FALLBACK=1 # Fallback to any available monitor when
 none matched

    export WESTON_OUTPUT_FLOW=horizontal # Laying outputs horizontally by default
    export WESTON_OUTPUT_FLOW=vertical # Laying outputs vertically by default
    export WESTON_OUTPUT_FLOW=same-as # Laying outputs at (0,0) by default
```

It also supports disabling the specified screen individually in the `output` section of weston.ini:

```
    # /etc/xdg/weston/weston.ini

    [output]
    name=LVDS-1

    mode=off
    # off|current|preferred|<WIDTHxHEIGHT@RATE>
```

For more specific screen position configuration, the dynamic configuration file can be used, for example:

```
    echo "output:HDMI-A-1:pos=100,200" > /tmp/.weston_drm.conf
```

2.12 Configuration of Input Devices

The Weston service requires at least one input device by default. If there is no input device, the special settings in the `core` section of weston.ini is needed:

```
# /etc/xdg/weston/weston.ini

[core]
require-input=false
```

If there are multiple screens in Weston, input devices should be bound to screens, you can configure it through the input device's udev rules's WL_OUTPUT env, such as:

```
# /lib/udev/rules.d/99-goodix-ts.rules
ATTRS{idVendor}=="dead", ATTRS{idProduct}=="beef", ENV{WL_OUTPUT}="HDMI-A-1"
```

Or WL_SEAT env, such as:

```
# /lib/udev/rules.d/99-goodix-ts.rules
ATTRS{idVendor}=="dead", ATTRS{idProduct}=="beef", ENV{WL_SEAT}="seat1"
```

```
# /etc/xdg/weston/weston.ini

[output]
name=LVDS-1

seat=seat1
```

2.13 Calibration of Input Devices

If you need to calibrate the touch screen, you can use WESTON_TOUCH_CALIBRATION environment, such as:

```
# /etc/profile.d/weston.sh
export WESTON_TOUCH_CALIBRATION="1.013788 0.0 -0.061495 0.0 1.332709 -0.276154"
```

The calibration parameters can be obtained by Weston calibration tool: weston-calibrator. After running this tool, a number of random points will be generated, and then click them in sequence to output the calibration parameters, such as: Final calibration values: 1.013788 0.0 -0.061495 0.0 1.332709 -0.276154

Or you can use the new Weston touch calibrator: weston-touch-calibrator, which needs:

```
# /etc/xdg/weston/weston.ini

[libinput]
touchscreen_calibrator=true
calibration_helper=/bin/weston-calibration-helper.sh
```

2.14 Configuration on the Platform without GPU

The Weston in the SDK uses GPU for render acceleration by default. For platforms without GPUs or GPU is not powerful enough, Rockchip RGA 2D acceleration can also be used instead.

The detailed configuration requires to enable BR2_PACKAGE_LINUX_RGA and BR2_PACKAGE_WESTON_DEFAULT_PIXMAN in the Buildroot SDK.

2.15 ARM AFBC modifier

For chips which support AFBC, the Weston in the SDK support using GPU based ARM AFBC compressed format for displaying.

```
# /etc/profile.d/weston.sh
# export WESTON_DISABLE_ATOMIC=1
export WESTON_ALLOW_GBM_MODIFIERS=1
```