

Rockchip Linux Qt WebEngine使用说明

文档标识: RK-SM-YF-324

发布版本: V1.0.0

日期: 2020-02-06

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

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

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

商标声明

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

本文档可能提及的其他所有注册商标或商标, 由其各自拥有者所有。

版权所有© 2020 瑞芯微电子股份有限公司

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

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: www.rock-chips.com

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: fae@rock-chips.com

前言

概述

文档主要介绍 Rockchip Linux Qt WebEngine使用说明，旨在帮助工程师更快上手Qt WebEngine开发及相关调试方法。

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

各芯片系统支持状态

芯片名称	Buildroot	Debian	Yocto
RK3288	Y	Y	N
RK3326/PX30	Y	Y	N
RK3328	Y	N	N
RK3399	Y	Y	N
RK3399PRO	Y	Y	N

修订记录

日期	版本	作者	修改说明
2020-02-06	V1.0.0	王晓腾	初始版本

目录

Rockchip Linux Qt WebEngine使用说明

1. Qt WebEngine
 - 1.1 概述
 - 1.2 架构
2. 不同系统Qt WebEngine的支持
 - 2.1 Buildroot
 - 2.2 Debian

1. Qt WebEngine

1.1 概述

Qt WebEngine模块提供了一个web浏览器,在不使用本地浏览器的情况下,它可以很容易地把Web内容嵌入到Qt应用程序中。

Qt WebEngine为渲染HTML, XHTML和SVG文档, 使用CSS和JavaScript, 提供了C++类和QML类型。

此文主要介绍Buildroot和Debian中Qt WebEngine的嵌入使用, 以及其调用从ffmpeg/mpp/vpu的multivideo硬解流程。

1.2 架构

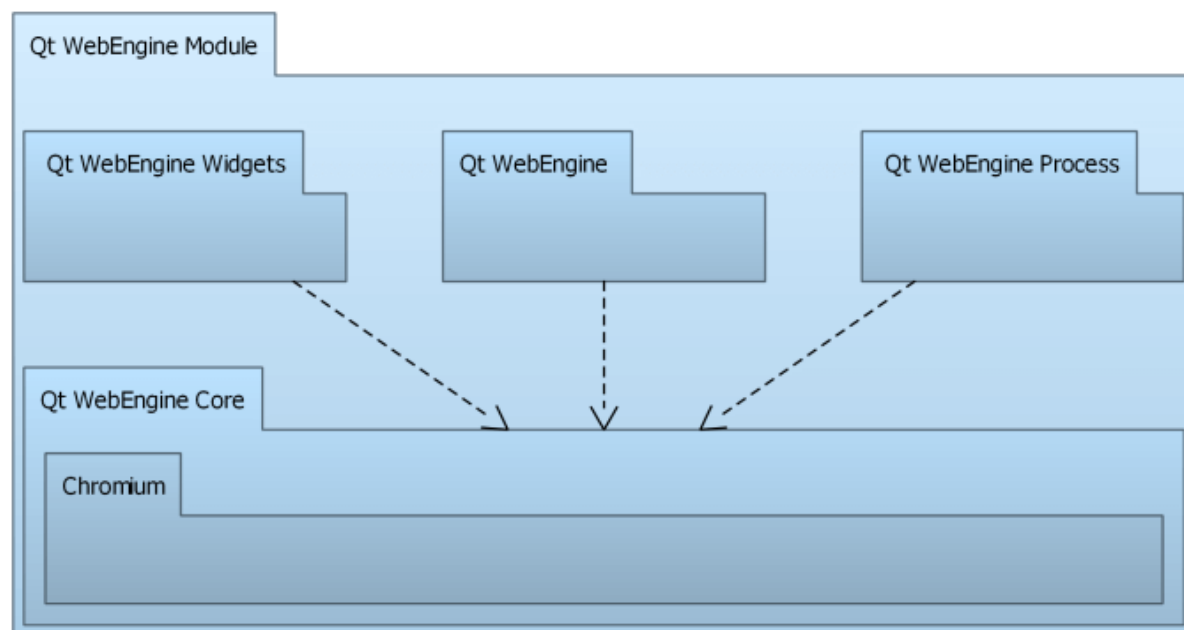
Qt WebEngine中的功能分为以下模块:

Qt WebEngine Widgets Module: 用于创建基于窗口小部件的web应用程序模块

Qt WebEngine Module: 用于创建基于Qt Quick的web应用程序模块

Qt WebEngine Core Module: 用于与Chromium交互的Qt-WebEngine核心模块

具体情况如下图所示:



更多详细内容可以参考[QT官方文档](#).

2. 不同系统Qt WebEngine的支持

2.1 Buildroot

Rockchip Linux 支持的Buildroot基于2018.02-rc3上开发，Qt WebEngine是基于5.12.2版本开发。

Buildroot支持Qt WebEngine需要打开config（BR2_PACKAGE_QT5WEBENGINE）以及相关配置，目前最新发布的SDK已经支持此功能，默认配置是关闭，需要打开如下配置：

```
#include "chromium.config"
```

如果需要ffmpeg实现视频硬解，也需要打开如下config：

```
#include "video_ffmpeg.config"
```

比如RK3399上支持Qt WebEngine功能，开启如下配置即可：

```
diff --git a/configs/rockchip_rk3399_defconfig
b/configs/rockchip_rk3399_defconfig
index dc84293..db6e177 100644
--- a/configs/rockchip_rk3399_defconfig
+++ b/configs/rockchip_rk3399_defconfig
@@ -1,8 +1,11 @@
#include "rk3399_arm64.config"
#include "base.config"
#include "base_extra.config"
+include "chromium.config"
#include "gpu.config"
#include "display.config"
+include "video_ffmpeg.config"
#include "video_mpp.config"
```

编译后Qt WebEngine的源码在 buildroot\$ vi output/rockchip_rk3399/build/qt5webengine-5.12.2/目录下，也可参考[QT官方源码](#)。

Buildroot中Qt WebEngine的配置设定可参考: buildroot\$ vi package/qt5/qt5webengine/目录，测试Demo在package/rockchip/rockchip_test/src/rockchip_test/chromium/目录下

```
#cat test_simplebrowser.sh
...
cd /usr/lib/qt/examples/webenginewidgets/simplebrowser
./simplebrowser --no-sandbox --disable-es3-gl-context
#./simplebrowser --no-sandbox --disable-es3-gl-context https://www.baidu.com
#./simplebrowser --no-sandbox --disable-es3-gl-context
"file:///oem/SampleVideo_1280x720_5mb.mp4"
#./simplebrowser --no-sandbox --disable-es3-gl-context --enable-logging --v=5
"file:///oem/SampleVideo_1280x720_5mb.mp4"
..
```

Buildroot编译时候已经指定了opengles，所以只需要处理前面的context问题，启动时候加参数--disable-es3-gl-context让chromium使用es2，因chromium视频硬解功能，需要访问一些设备节点，所以启动simplebrowser需要加--no-sandbox参数。

2.2 Debian

Rockchip Linux Debian 9 (stretch) 官方Qt为5.7, 不支持WebEngine, SDK中使用的Qt相关包是基于buster源更新, 版本为5.11, 所以手动安装时候需要修改源。

```
export DISPLAY=:0
su linaro -c "xhost +"
echo "deb http://ftp.cn.debian.org/debian buster main" >> /etc/apt/sources.list
apt-get update
apt-get install qtwebengine5-examples
/usr/lib/aarch64-linux-gnu/qt5/examples/webengine/minimal/minimal --no-sandbox
```

测试后把/etc/apt/sources.list修改还原。

如果是移植QT官方WebEngine编译, 比如5.12.2有如下注意项:

- Debian官方的qtwebengine是编译了xcb glx (RK平台为mesa软件实现)和xcb egl(mali gpu实现),优先走glx.
这个需要通过环境变量设置使用egl,否则用软件渲染 (和关闭RGA效果一样);

```
export QT_XCB_GL_INTEGRATION=xcb_egl
```

- 走egl后出现"Cannot find EGLConfig, returning null config".这是因为Qt的xcb实现中, default renderable type设置为opengl (mali库不支持).这个问题可以:
应用端通过QSurfaceFormat的setRenderableType更改默认设置为QSurfaceFormat::OpenGLES (具体请google该关键字),
也可以参考官方demo: qt5base-5.12.2# vi examples/opengl/computeegles31/main.cpp
或配置里面去掉opengl (只保留opengles)重编QT, 默认使用opengles.

```
qt5base-5.12.2# git diff src/platformsupport/eglconvenience/qeglconvenience.cpp
diff --git a/src/platformsupport/eglconvenience/qeglconvenience.cpp
b/src/platformsupport/eglconvenience/qeglconvenience.cpp
index 020d035..a4156cb 100644
--- a/src/platformsupport/eglconvenience/qeglconvenience.cpp
+++ b/src/platformsupport/eglconvenience/qeglconvenience.cpp
@@ -252,7 +252,7 @@ EGLConfig QEglConfigChooser::chooseConfig()
     break;
     #ifdef EGL_VERSION_1_4
         case QSurfaceFormat::DefaultRenderableType:
-#ifndef QT_NO_OPENGL
+#if 0//ndef QT_NO_OPENGL
             if (QOpenGLContext::openGLModuleType() == QOpenGLContext::LibGL)
                 configureAttributes.append(EGL_OPENGL_BIT);
             else
```

- 修改后出现看到的 "eglCreateContext failed with error EGL_BAD_CONTEXT", 这个是因为Qt默认创建的是es2的context, 然后chromium里面尝试封装成es3导致出错. 这个问题可以:
启动时候加参数--disable-es3-gl-context让chromium使用es2或应用端通过QSurfaceFormat setVersion更改默认设置为3 (具体请google该关键字), 也可以参考官方demo:
qt5base-5.12.2# vi examples/opengl/computeegles31/main.cpp或修改qt xcb插件, 让Qt创建es3 context:

```
qt5base-5.12.2# git diff
src/plugins/platforms/xcb/gl_integrations/xcb_egl/qxcbeglintegration.cpp
```

```

diff --git
a/src/plugins/platforms/xcb/gl_integrations/xcb_egl/qxcbeglintegration.cpp
b/src/plugins/platforms/xcb/gl_integrations/xcb_egl/qxcbeglintegration.cpp
index fe18bc2..bb8c72c 100644
--- a/src/plugins/platforms/xcb/gl_integrations/xcb_egl/qxcbeglintegration.cpp
+++ b/src/plugins/platforms/xcb/gl_integrations/xcb_egl/qxcbeglintegration.cpp
@@ QXcbWindow *QXcbEglIntegration::createWindow(QWindow *window) const
    QPlatformOpenGLContext *QXcbEglIntegration::createPlatformOpenGLContext
    (QOpenGLContext *context) const
    {
        QXcbScreen *screen = static_cast<QXcbScreen *>(context->screen()->handle());
-       QXcbEglContext *platformContext = new QXcbEglContext(screen-
>surfaceFormatFor(context->format()),
+
+       QSurfaceFormat format = screen->surfaceFormatFor(context->format());
+       format.setMajorVersion(3);
+
+       QXcbEglContext *platformContext = new QXcbEglContext(format,
                                                                    context->shareHandle(),
                                                                    eglDisplay(),

```

- 最后出现“Failed to initialize extensions”

是因为webengine同时链接opengl(mesa)、opengles(mali)库里面一些符号表存在冲突，导致一部分使用mesa，一部分mali。此问题可以在编译应用时候添加libGLv2.so库依赖，这样会优先绑定其符号表。测试时可以直接用工具修改应用添加依赖，如：patchelf --add-needed libGLv2.so minimal。