

# RK3308 显示 开发指南

---

发布版本：1.0

作者邮箱：[jkand.huang@rock-chips.com](mailto:jkand.huang@rock-chips.com)

日期：2019.04

文件密级：公开资料

---

## 前言

### 概述

本文档旨在指导工程师如何在Rockchip RK3308平台上使用显示，并进行二次开发。

### 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

### 产品版本

芯片名称	内核版本
RK3308	4.4

### 修订记录

日期	版本	作者	修改说明
2019-04-15	V1.0	黄开辉	初始版本

---

## RK3308 显示 开发指南

[U-Boot配置](#)

[Kernel 配置](#)

[BuildRoot 配置](#)

[menuconfig 配置](#)

[编译QT](#)

[环境变量设置](#)

[编译脚本使用](#)

---

## U-Boot配置

---

u-boot 打开如下配置，会将u-boot log 显示出来

```
CONFIG_DM_VIDEO=y
CONFIG_DISPLAY=y
CONFIG_DRM_ROCKCHIP=y
CONFIG_DRM_ROCKCHIP_RGB=y
```

## Kernel 配置

kernel 打开如下配置，会将kernel log 显示出来

```
CONFIG_DRM=y
CONFIG_DRM_ROCKCHIP=y
CONFIG_ROCKCHIP_RGB=y
CONFIG_DRM_PANEL_SIMPLE=y
CONFIG_BACKLIGHT_LCD_SUPPORT=y
CONFIG_BACKLIGHT_CLASS_DEVICE=y
CONFIG_BACKLIGHT_PWM=y
```

dts 参考 rk3308-evb-ext-v10.dtsi 文件

注意：RK3308 只支持RGB 和 MCU 两种显示接口

## BuildRoot 配置

按照以下步骤，即可在buildroot/output/xxxxxx/target/usr/lib/qt/examples/ 目录下生成QT Demo。

### menuconfig 配置

```
BR2_PACKAGE_QT5=y
BR2_PACKAGE_QT5BASE_EXAMPLES=y //如不需要Demo，该选项可以去掉，减少固件大小
BR2_PACKAGE_QT5BASE_WIDGETS=y
+BR2_PACKAGE_QT5BASE_FONTCONFIG=y
BR2_PACKAGE_QT5BASE_GIF=y
BR2_PACKAGE_QT5BASE_JPEG=y
BR2_PACKAGE_QT5BASE_PNG=y
BR2_PACKAGE_QT5QUICKCONTROLS=y
BR2_PACKAGE_QT5QUICKCONTROLS2=y
BR2_PACKAGE_QT5BASE_FONTCONFIG=y
BR2_PACKAGE_LIBDRM=y
BR2_PACKAGE_LIBDRM_INSTALL_TESTS=y
BR2_PACKAGE_DEJAVU=y //字体库
```

### 编译QT

1. make qt5base-dirclean
2. make qt5base -j32
3. make -j32

4. ./mkfirmware.sh

## 环境变量设置

以usr/lib/qt/examples/quickcontrols2/wearable/wearable 为例，环境变量

```
export QT_QPA_FB_DRM=1 //使用drm
export QT_QPA_PLATFORM=linuxfb //使用linuxfb 框架
export QMLSCENE_DEVICE=softwarecontext //使用CPU 绘制
```

## 编译脚本使用

- 1.将工程放到external 目录下
- 2.将以下脚本放到工程根目录
- 3.修改脚本QMAKE, STRIP, BUILDROOT\_TARGET\_PATH到对应的目录

```
#!/bin/sh
QT_PROJECT_FILE=wearable.pro #工程pro文件
TOP_DIR=$(pwd)
APP_DIR_NAME=$(basename $TOP_DIR)
BUILDROOT_TARGET_PATH=$(pwd)/../../buildroot/output/rockchip_rk3308_32_release/target/
TARGET_APP_PATH=$BUILDROOT_TARGET_PATH/usr/local/$APP_DIR_NAME/
QMAKE=$(pwd)/../../buildroot/output/rockchip_rk3308_32_release/host/bin/qmake
STRIP=$(pwd)/../../buildroot/output/rockchip_rk3308_32_release/host/usr/bin/arm-linux-strip
TARGET_EXECUTABLE=""

#get parameter for "-j2~8 and clean"
result=$(echo "$1" | grep -Eo '*clean')
if [ "$result" = "" ];then
    mulcore_cmd=$(echo "$1" | grep '^-j[0-9]\{1,2\}$')
elif [ "$1" = "clean" ];then
    make clean
elif [ "$1" = "distclean" ];then
    make clean
else
    mulcore_cmd=-j16
fi

#qmake and build target
$QMAKE
make $mulcore_cmd
rm -rf $TARGET_APP_PATH
if [ ! -d "$TARGET_APP_PATH" ]; then
    mkdir -p "$TARGET_APP_PATH"
fi
for file in .//*
do
    if test -x $file
```

```
then
    elf=$(file -e elf $file)
    result=$(echo $elf | grep "LSB executable")
    if [ -n "$result" ]
    then
        TARGET_EXECUTABLE=$(basename "$file")
        echo "found executable file $TARGET_EXECUTABLE"
        $STRIP $TOP_DIR/$TARGET_EXECUTABLE
        cp $TOP_DIR/$TARGET_EXECUTABLE $TARGET_APP_PATH
        #cp -r $TOP_DIR/resources $TARGET_APP_PATH
        echo "$TARGET_EXECUTABLE app is ready."
    fi
fi
done

make clean

#call just for buid_all.sh
if [ "$1" = "cleanthen" ] || [ "$2" = "cleanthen" ];then
    make clean
fi
```