

密级状态：绝密() 秘密() 内部资料() 公开(✓)

rk3399 ubuntu16.04 开发说明

(技术部，第二系统产品部)

文件状态： [] 草稿 [] 正式发布 [✓] 正在修改	文件标识：	rk3399 ubuntu16.04 开发说明
	当前版本：	V1.0
	作 者：	黄国椿 蓝斌元
	完成日期：	2018-01-15
	审 核：	蓝斌元
	完成日期：	2018-01-15

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co . , Ltd

(版本所有,翻版必究)

文档修改记录:

日期	修订版本	修订内容	修改人	核定人
2017-09-19	V1.00	第一版发布	黄国椿	蓝斌元
2018-01-15	V1.10	完善 wifi、gpu、vpu、qt 等功能	黄国椿	蓝斌元

目录

1 下载和解压 ubuntu.....	1
2 安装 qemu.....	1
3 准备网络.....	2
4 进入根文件系统进行操作.....	2
5 安装 xubuntu.....	2
6 添加用户及设置密码.....	3
7 扩展文件系统大小.....	4
8 QT 源码编译.....	5
9 gpu 安装.....	6
10 gstreamer-rockchip & mpp deb 编译和安装.....	7
11 制作根文件系统.....	9
12 桌面环境下测试 vpu 和 gpu.....	9
附件.....	10

1 下载和解压 ubuntu

rk3399-excavator ubuntu 根文件系统是基于 ubuntu-base-16.04 来创建的。ubuntu 16.04 下载地址：

```
http://cdimage.ubuntu.com/ubuntu-base/releases/16.04/release/
```

在这里我们选择 32 位或者 64 位的包（rk3399 建议使用 64 位的）：

```
ubuntu-base-16.04-core-armhf.tar.gz
```

```
ubuntu-base-16.04-core-arm64.tar.gz
```

下载完之后，创建临时文件夹并解压根文件系统：

```
mkdir ubuntu
```

```
sudo tar -xpf ubuntu-base-16.04-core-armxx.tar.gz -C ubuntu
```

2 安装 qemu

在 Linux PC 主机上安装模拟器

```
sudo apt-get install qemu-user-static
```

拷贝 qemu

32 位的 ubuntu:

```
sudo cp /usr/bin/qemu-arm-static ubuntu/usr/bin
```

64 位的 ubuntu:

```
sudo cp /usr/bin/qemu-aarch64-static ubuntu/usr/bin
```

3 准备网络

拷贝主机的 dns 到根文件系统

```
sudo cp -b /etc/resolv.conf ubuntu/etc/resolv.conf
```

4 进入根文件系统进行操作

以临时文件为根目录

```
sudo chroot ubuntu
```

修改 /etc/apt/sources.list, 把被注释的源重新打开

```
sed -i -e 's,^# deb\(.*\)$,deb\1,g' /etc/apt/sources.list
```

更新源索引并升级已经安装的包

```
apt update
```

```
apt upgrade
```

预安装功能

```
apt install vim git net-tools ubuntu-advantage-tools onboard
```

```
glmark2-es2
```

5 安装 xubuntu

需要考虑固件大小的话建议使用轻量级的桌面服务系统:

```
apt install xubuntu-core
```

无需考虑固件大小的话可以选择如下桌面服务系统(集成火狐浏览器和其他插件):

```
apt install xubuntu-desktop
```

6 添加用户及设置密码

添加用户:

```
useradd -s '/bin/bash' -m -G adm,sudo rockchip
```

给用户设置密码:

```
passwd rockchip
```

给 root 用户设置密码:

```
passwd root
```

若不需要桌面登陆界面, 直接用户登录, 可以将创建用户列入自动登录名单:

```
echo "autologin-user=rockchip" >
```

```
/usr/share/lightdm/lightdm.conf.d/60-xubuntu.conf
```

若想无需密码自动登陆 ubuntu 的串口终端, 进行如下修改以 root 或者用户 rockchip 登陆:

```
vi lib/systemd/system/serial-getty@.service
```

```
[Service]
ExecStart=-/sbin/agetty --keep-baud -a root 115200,38400,9600 %I $TERM
Type=idle
Restart=always
```

```
[Service]
ExecStart=-/sbin/agetty --keep-baud -a rockchip 115200,38400,9600 %I $TERM
Type=idle
Restart=always
```

7 扩展文件系统大小

添加相应 service 的 unit 文件

添加文件: /lib/systemd/system/resize-helper.service

```
[Unit]
Description=Resize root filesystem to fit available disk space
After=systemd-remount-fs.service
[Service]
Type=oneshot
ExecStart=/usr/sbin/resize-helper
ExecStartPost=/bin/systemctl disable resize-helper.service
[Install]
WantedBy=basic.target
```

以上文件指定该服务在 systemd-remount-fs.service 服务之后启动，且服务启动完成之后就退出不产生进程，ExecStart 定义了需要添加的服务主体内容：

/usr/sbin/resize-helper

```
# we must be root
[ $(whoami) = "root" ] || { echo "E: You must be root" && exit 1; }
# we must have few tools
SGDISK=$(which sgdisk) || { echo "E: You must have sgdisk" && exit 1; }
PARTED=$(which parted) || { echo "E: You must have parted" && exit 1; }
PARTPROBE=$(which partprobe) || { echo "E: You must have partprobe" && exit 1; }
RESIZE2FS=$(which resize2fs) || { echo "E: You must have resize2fs" && exit 1; }
# find root device
ROOT_DEVICE=$(findmnt --noheadings --output=SOURCE / | cut -d' ' -f1)
# prune root device (for example UUID)
ROOT_DEVICE=$(realpath ${ROOT_DEVICE})
# get the partition number and type
PART_ENTRY_NUMBER=$(udevadm info --query=property --name=${ROOT_DEVICE} | grep '^ID_PART_ENTRY_NUMBER=' | cut -d'=' -f2)
PART_TABLE_TYPE=$(udevadm info --query=property --name=${ROOT_DEVICE} | grep '^ID_PART_TABLE_TYPE=' | cut -d'=' -f2)
# find the block device
DEVICE=$(udevadm info --query=path --name=${ROOT_DEVICE} | awk -F'/' '{print $(NF-1)}')
DEVICE="/dev/${DEVICE}"
```

```
if [ "$PART_TABLE_TYPE" = "gpt" ]; then
    ${SGDISK} -e ${DEVICE}
    ${PARTPROBE}
fi
${PARTED} ${DEVICE} resizepart ${PART_ENTRY_NUMBER} 100%
${PARTPROBE}
${RESIZE2FS} "${ROOT_DEVICE}"
```

注：resize-helper 需要添加可执行权限。

8 QT 源码编译

下载 qt 源码 [qt-everywhere-opensource-src-5.9.3.tar.xz](http://mirror.bit.edu.cn/qtproject/archive/qt/5.9/5.9.3/sin):

<http://mirror.bit.edu.cn/qtproject/archive/qt/5.9/5.9.3/sin>
[gle/](http://mirror.bit.edu.cn/qtproject/archive/qt/5.9/5.9.3/sin)

解压

```
xz -d qt-everywhere-opensource-src-5.9.3.tar.xz
tar xvf qt-everywhere-opensource-src-5.9.3.tar
```

configure 配置

```
./configure -confirm-license -opensource -release -make libs -linuxfb
-xcb -pch -platform linux-aarch64-gnu-g++ -make examples -nomake tools
-nomake tests -prefix /usr/lib/aarch64-linux-gnu/qt5
```

编译

```
make -j8
make install
```

最终 qt 将编译输出到 configure -prefix 选项所指定的目录下。

9 gpu 安装

以下用.deb 格式的包对 ubuntu 进行补充，deb 格式的包可以通过 docker 来构建，docker 的源码和环境准备工作可以参考（github 上的源码默认是 32 位，64 位 docker 需要修改 dockerfile:armhf->arm64）：

<https://github.com/rockchip-linux/docker-rockchip>

下载 libmali 源码：

```
git clone https://github.com/rockchip-linux/libmali.git
```

下载 xserver 源码：

```
git clone https://github.com/rockchip-linux/xserver.git
```

以上编译之后可以生成对应的.deb 包，所需的.deb 文件可以在附件的百度云地址或联系我们获取，按如下顺序安装：

```
cd rk3399_gpu/libmali/  
dpkg -i libmali-rk-dev_1.5-4_arm64.deb  
dpkg -i libmali-rk-midgard-4th-r13p0_1.5-4_arm64.deb  
update-alternatives --remove aarch64-linux-gnu_egl_conf /usr/lib/a  
arch64-linux-gnu/mesa-egl/ld.so.conf  
ldconfig
```

```
cd rk3399_gpu/xserver/  
dpkg -i xserver-common_1.18.4-1_all.deb  
dpkg -i libcrypt20_1.7.2-2ubuntu1_arm64.deb  
dpkg -i xserver-xorg-core_1.18.4-1_arm64.deb
```

添加相关 udev 文件

```
vi /lib/udev/rules.d/50-mali.rules  
KERNEL=="mali0", MODE="0666", GROUP="video"
```

10 gstreamer-rockchip & mpp deb 编译和安装

下载 gstreamer-rockchip 源码:

```
git clone https://github.com/rockchip-linux/gstreamer-rockchip.git
```

编译前需要打上如下补丁:

```
https://github.com/rockchip-linux/rk-rootfs-build/tree/master/packages-  
patches/gstreamer-rockchip/release_20171218
```

下载 mpp 源码:

```
git clone https://github.com/rockchip-linux/mpp.git
```

编译前需要打上如下补丁:

```
https://github.com/rockchip-linux/rk-rootfs-build/tree/master/packages-  
patches/mpp/release\_20171218
```

(0002-DEBIAN-ONLY-use-mmap64.patch 除外)

下载 gstreamer-rockchip-extra 源码:

```
git clone https://github.com/rockchip-linux/gstreamer-rockchip-extra.git
```

以上通过 docker 编译后的.deb 包如下:

```
hgc@rk-hgc: /media/hgc/TOSHIBA/rk3399_vpu/rkmpp&gstream$ tree  
├── gstreamer1.0-rockchip1_20171218-1_arm64.deb  
├── gstreamer1.0-rockchip1-dbgsym_20171218-1_arm64.deb  
├── gstreamer1.0-rockchip1-extra_20171218-1_arm64.deb  
├── gstreamer1.0-rockchip1-extra-dbgsym_20171218-1_arm64.deb  
├── librockchip-mpp1_20171218-1_arm64.deb  
├── librockchip-mpp1-dbgsym_20171218-1_arm64.deb  
├── librockchip-mpp-dev_20171218-1_arm64.deb  
├── librockchip-mpp-static_20171218-1_arm64.deb  
├── librockchip-vpu0_20171218-1_arm64.deb  
├── librockchip-vpu0-dbgsym_20171218-1_arm64.deb  
├── rockchip-mpp-demos_20171218-1_arm64.deb  
└── rockchip-mpp-demos-dbgsym_20171218-1_arm64.deb
```

另外还需到 ubuntu 官网准备构建 gstreamer 环境：

```
/media/hgc/TOSHIBA/rk3399_vpu/gstream_env/
├── gir1.2-gst-plugins-base-1.0_1.12.2-1_arm64.deb
├── gir1.2-gstreamer-1.0_1.12.2-1_arm64.deb
├── gstreamer1.0-alsa_1.12.2-1_arm64.deb
├── gstreamer1.0-plugins-base_1.12.2-1_arm64.deb
├── gstreamer1.0-plugins-good_1.12.2-1ubuntu1_arm64.deb
├── gstreamer1.0-pulseaudio_1.12.2-1ubuntu1_arm64.deb
├── gstreamer1.0-tools_1.12.2-1_arm64.deb
├── gstreamer1.0-x_1.12.2-1_arm64.deb
├── libglib2.0-dev_2.48.2-0ubuntu1_arm64.deb
├── libgstreamer1.0-0_1.12.2-1_arm64.deb
├── libgstreamer1.0-0-dbg_1.12.2-1_arm64.deb
├── libgstreamer1.0-dev_1.12.2-1_arm64.deb
├── libgstreamer-plugins-base1.0-0_1.12.2-1_arm64.deb
├── libgstreamer-plugins-base1.0-dev_1.12.2-1_arm64.deb
├── libgstreamer-plugins-good1.0-0_1.12.2-1ubuntu1_arm64.deb
├── libgstreamer-plugins-good1.0-dev_1.12.2-1ubuntu1_arm64.deb
├── libvpx4_1.6.1-3_arm64.deb
└── plugin_bad
    ├── gir1.2-gst-plugins-bad-1.0_1.12.2-1ubuntu2_arm64.deb
    ├── gstreamer1.0-plugins-bad_1.12.2-1ubuntu2_arm64.deb
    ├── libgstreamer-plugins-bad1.0-0_1.12.2-1ubuntu2_arm64.deb
    └── libgstreamer-plugins-bad1.0-dev_1.12.2-1ubuntu2_arm64.deb
```

以上所需的 deb 文件可以在附件的百度云地址或联系我们获取。

按如下安装：

```
$cd ~/rk3399_vpu/gstream_env
```

```
$sudo dpkg -i *.deb
```

以上正常安装，如果有依赖问题， apt-get -f install 解决。

```
$cd ~/rk3399_vpu/rkmpp&gstream
```

```
$sudo dpkg -i *.deb
```

以上正常安装，如果有依赖问题， apt-get -f install 解决。

```
$cd ~/rk3399_vpu/gstream_env/plugin_bad
```

```
$sudo dpkg -i *.deb
```

如果这一步安装出现依赖问题， 可以先不处理,保持原状。因为 ubuntu16.04 默认的源 gstreamer 都是 1.8 的，现在安装的是 1.12 的，有一些插件的依赖，目前很难解决。但是这些插件我们用不上。

以上修改完成根文件系统后就可以退出了。

```
exit
```

11 制作根文件系统

制作自己的根文件系统，大小依据自己的根文件系统而定，注意依据 ubuntu 文件夹的大小来修改 count 值：

```
dd if=/dev/zero of=linuxroot.img bs=1M count=4096
sudo mkfs.ext4 linuxroot.img
mkdir rootfs
sudo mount linuxroot.img rootfs/
sudo cp -rfp ubuntu/* rootfs/
sudo umount rootfs/
e2fsck -p -f linuxroot.img
resize2fs -M linuxroot.img
```

12 桌面环境下测试 vpu 和 gpu

vpu:

```
sudo gst-launch-1.0 uridecodebin uri=file:///home/firefly/test.mp4 !
```

```
rkximagesink
```

或

```
gst-launch-1.0 filesrc location=/mnt/udisk/lg_4k.mp4 ! qtdemux !
```

```
h265parse ! mppvideodec ! rkximagesink
```

或

直接启动桌面 qt_player 的 qt 播放器播放测试。

gpu:

```
glmark2-es2
```

附件

GPU 和 VPU 所需 deb 文件可从如下地址获取：

百度云盘

链接：<https://pan.baidu.com/s/1kXnDt0j> 密码：x5pi