CNXSoft – Embedded Software Development

Software development cycle, linux programming, hardware development....

Getting Started with Raxda Rock – How to Generate and Flash Ubuntu Saucy Server and Desktop Images

January 27th, 2014 cnxsoft Leave a comment Go to comments

I've already written a guide showing how to build and install Android in Radxa Rock. Today I'm going to build the Linux kernel, generate a Ubuntu server images based latest Ubuntu Linaro server release, flash the image to the NAND flash, and show how to install LXDE or XFCE desktop environment. If you are just interested in installing one of the latest supported images, you can download Android, dual boot (Android/Ubuntu), Ubuntu ALIP, and Router images for Radxa Rock

@ http://dl.radxa.com/rock/images/, and skip “build” and “generate” instructions, and just follow the flashing instructions below. All steps in this tutorial will be done in Ubuntu 13.10, and lots of the instructions below follow hwswbits blog and Ubuntu Radxa Wiki.

Build the Linux kernel

Let’s start by building the Linux kernel. I’m assuming you’ve already install development tools in Ubuntu (e.g. apt-get install build-essentials). You can refer to the list at the beginning of the Radxa Rock Android post for packages to install with apt-get.

Let’s create a working directory, get the code, and build the kernel.

mkdir radxa_ubuntu
cd radxa_ubuntu

```
git clone -b wip/rockchip-3.0-radxa-rock https://github.com/linux-rockchip/rockchip-3.0.git
cd rockchip-3.0
export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabihf-
make rk3188_radxa_rock_defconfig
make -j8
```

At this stage, we’ve build the kernel. However, if we use it as is, DNS and some other network functions will not work because CONFIG_ANDROID_PARANOID is enabled in the default config file. Run *make arch=ARM menuconfig,* and go to *Networking support -> Networking options* to disable *Only allow certain groups to create sockets.* Run make again:

```
make -j8
```

Let’s complete this section, by building the kernel modules:

```
mkdir modules
export INSTALL_MOD_PATH=./modules
make modules && make modules_install
cd ..
```

Generate Initrd

The initrd is used at boot time to speed up boot among other things.

```
git clone https://github.com/radxa/initrd.git
cd initrd
find . ! -path "./.git*"  | cpio -H newc  -ov > ../initrd.img
cd ..
```

Generate Boot.img

In order to create boot.img which will be flashed in the boot partition, we need to download mkbootimg tool, and run the following commands:

```
wget http://dl.radxa.com/rock/tools/linux/mkbootimg
sudo apt-get install lib32stdc++6
chmod +x mkbootimg
./mkbootimg --kernel rockchip-3.0/arch/arm/boot/Image --ramdisk initrd.img -o boot.img
```

That's all good as we should now have a working boot.img. ~~I'll upload my boot.img soon.~~ You can download boot-radxa-20140127.img.xz.

Generate Ubuntu (Linaro) Server Rootfs

We will first create an empty 1GB file (rock_rootfs.img), format it to EXT-4, and mount it as a loop device:

```
dd if=/dev/zero of=rock_rootfs.img bs=1M count=1024
mkfs.ext4 -F -L linuxroot rock_rootfs.img
mkdir mnt
sudo mount -o loop rock_rootfs.img ./mnt
```

I'll download and extract the latest Ubuntu Saucy server image from Linaro available as today:

```
wget https://releases.linaro.org/13.12/ubuntu/saucy-images/server/linaro-saucy-server-20131216-586.ta
sudo tar zxvf linaro-saucy-server-20131216-586.tar.gz -C ./mnt
cd ./mnt
sudo mv binary/* .
sudo rm binary -fr
cd ..
```

Now we need to copy the kernel modules and firmware to our rootfs:

```
sudo mkdir ./mnt/lib/firmware
sudo cp -r rockchip-3.0/modules/lib/modules/3.0.36+/ ./mnt/lib/modules
sudo mkdir ./mnt/lib/firmware
sudo cp -r rockchip-3.0/firmware/* ./mnt/lib/firmware/
```

I'd like Ethernet to work at boot time with DHCP so I edit the relevant file:

```
sudo vi ./mnt/etc/network/interfaces
```

and add the two lines:

```
auto eth0
iface eth0 inet dhcp
```

If you've already referred to Radxa Rock Ubuntu wiki, at this stage they use qemu and chroot to install extra package, I'll skip that part, because it can be done inside the board instead, and I came across some error with qemu when accessing the network:

```
apt-get update
0% [Working]
qemu: Unsupported syscall: 374
qemu: Unsupported syscall: 374
Err http://ports.ubuntu.com saucy
```

We still have one more step to create and enable a script to automatically detect the rootfs partition at boot time.

```
sudo vi ./mnt/usr/local/bin/mtd-by-name.sh
```

And copy the following to the file:

```
#!/bin/sh -e
# mtd-by-name link the mtdblock to name
# radxa.com, thanks to naobsd
rm -rf /dev/block/mtd/by-name/
mkdir -p /dev/block/mtd/by-name
for i in `ls -d /sys/class/mtd/mtd*[0-9]`; do
    name=`cat $i/name`
    tmp="`echo $i | sed -e 's/mtd/mtdblock/g'`"
    dev="`echo $tmp |sed -e 's/\/sys\/class\/mtdblock/\/dev/g'`"
    ln -s $dev /dev/block/mtd/by-name/$name
done
```

Make sure the script is executable and edit one of the startup script:

```
sudo chmod +x ./mnt/usr/local/bin/mtd-by-name.shsudo
sudo vi ./mnt/etc/rc.local
```

And add the line below before exit 0:

```
/usr/local/bin/mtd-by-name.sh
```

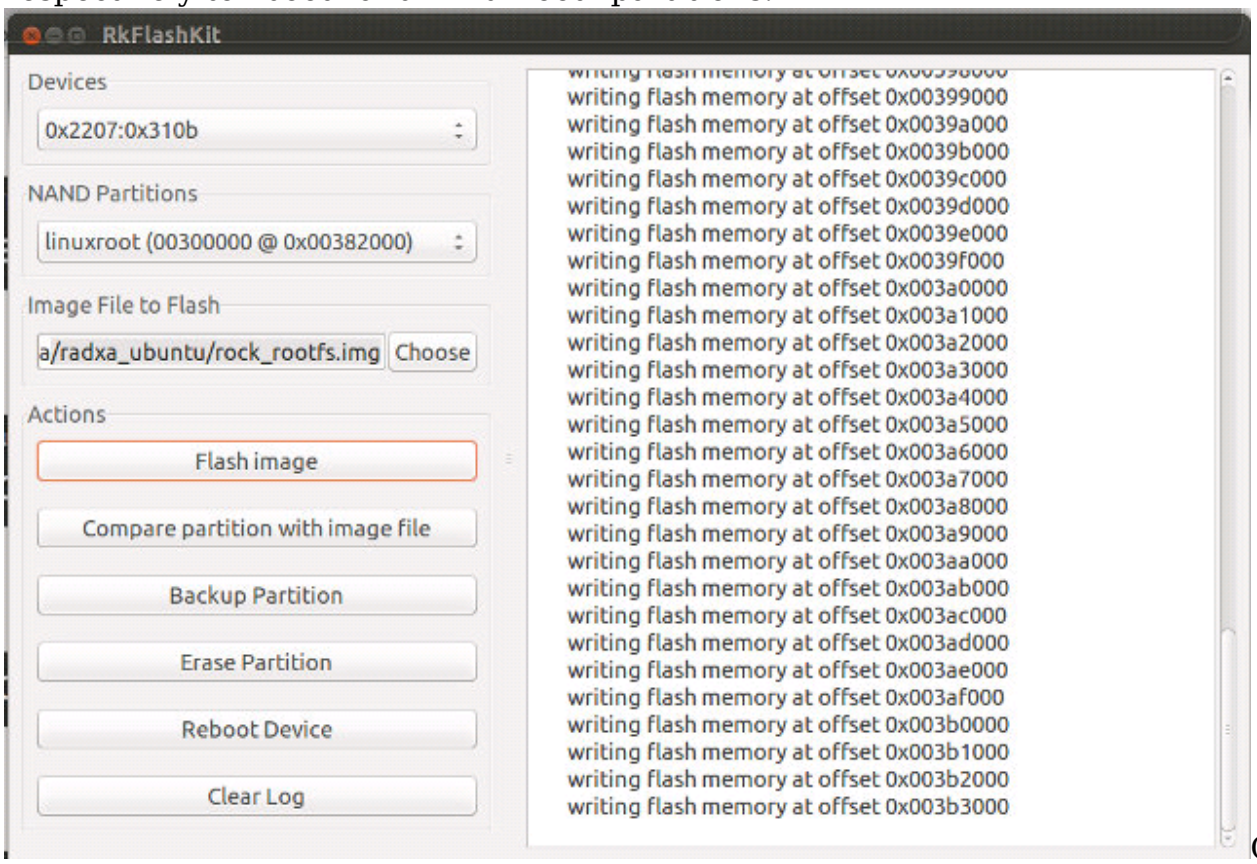
We are done, let's umount the rootfs:

```
sudo umount ./mnt/
```

~~I'll upload my Ubuntu Saucy Server rootfs soon.~~ Download link for Ubuntu Saucy server rootfs: radxa_rock_ubuntu_server_saucy_20140127.img.xz (95.4 MB)

Flashing Ubuntu to Radxa Rock

Now that we've got boot.img and rock_rootfs.img, we are ready to flash Ubuntu to the board. If you've haven't installed it already, install RKFlashKit, press the recovery button on Radxa Rock, start RkFlashKit, and flash boot.img and rock_rootfs.img respectively to "boot" and "linuxroot" partitions.



Click on Reboot device, and you should be able to access the console via HDMI or the serial console.

Ubuntu Saucy Server Available RAM and storage

SSH is not installed, but you can call install dropbear or ssh if you want to access the board remotely:

```
apt-get update
apt-get install dropbear
```

You can now SSH the board with linaro user using the password: linaro.

We've got plenty of available RAM:

```
free -mh
             total       used       free     shared    buffers     cached
Mem:          1.8G       228M       1.6G          0B       7.9M       170M
-/+ buffers/cache:        49M       1.8G
Swap:           0B         0B         0B
```

But there's only a 1GB flash partition usable with 409MB available.

```
df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mtdblock9 1008M  548M  409M  58% /
udev            924M   68K  924M   1% /dev
none            4.0K     0  4.0K   0% /sys/fs/cgroup
none            186M  264K  185M   1% /run
none            5.0M     0  5.0M   0% /run/lock
none            927M     0  927M   0% /run/shm
none            100M     0  100M   0% /run/user
```

However, there's also a 4GB "user" partition:

```
cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00400000 00004000 "misc"
mtd1: 00800000 00004000 "kernel"
mtd2: 01000000 00004000 "boot"
mtd3: 02000000 00004000 "recovery"
mtd4: 04000000 00004000 "backup"
mtd5: 08000000 00004000 "cache"
mtd6: 40000000 00004000 "userdata"
mtd7: 00400000 00004000 "kpanic"
mtd8: 20000000 00004000 "system"
mtd9: 60000000 00004000 "linuxroot"
mtd10: 10c000000 00004000 "user"
```

We can mount it as follows:

```
sudo mkdir /mnt/user
sudo mount /dev/mtdblock10 /mnt/user
```

And get some extra space:

```
df -h
Filesystem       Size  Used Avail Use% Mounted on
/dev/mtdblock9  1008M  548M  409M  58% /
...
/dev/mtdblock10  4.2G  8.0K  4.2G   1% /mnt/user
```

But we'd still only get 5GB out of a 8GB NAND flash, and there's not enough space on the rootfs to install a desktop environment.

Increasing Radxa Rock Ubuntu Rootfs Partition Size

Let's see what can be done to get rid off useless Android partition, and get a 7GB rootfs.

First we need to modify the Android parameter file to change the MTD layout for linux (parameter-linux). Refer to the Android post to get the parameter files in rockdev directory.

```
cp ../radxa_rock_android/rockdev/parameter parameter-linux
vi parameter-linux
```

Edit parameter-linux as follow with only 2 partitions for boot.img and rock_rootfs.img (linuxroot):

```
FIRMWARE_VER:4.2.2
MACHINE_MODEL:radxa_rock
MACHINE_ID:007
MANUFACTURER:RADXA
MAGIC: 0x5041524B
ATAG: 0x60000800
MACHINE: 3066
CHECK_MASK: 0x80
KERNEL_IMG: 0x60408000
#RECOVER_KEY: 1,1,0,20,0
CMDLINE:console=ttyFIQ0 console=tty0 root=/dev/block/mtd/by-name/linuxroot rw
rootfstype=ext4 init=/sbin/init initrd=0x62000000,0x00800000
mtdparts=rk29xxnand:0x00008000@0x00002000(boot),0x00E00000@0x00A000(linuxroot)
```

Now get the upgrade_tool (download link), enter recovery mode and flash the parameter file:

```
./upgrade_tool di -p parameter-linux
```

Start RkFlashKit again to flash boot.img and rock_rootfs.img again. This time, you should only see two partitions in the tools: boot and linuxroot.

However, the first time I tried to flash rock_rootfs.img, I got the following error message in RkFlashKit:

```
"Flash memory at 0x0000a000 is differnt from file!"
```

And after reboot, in the serial console, I got quite a few error message similar to:

```
FlashReadRetry error!!,row = 0
```

A low level format, before using RkFlashKit, and flashing the parameter file fixed the issue:

```
./upgrade_tool lf
```

You may want to run the command above first in any case, because flashing rock_rootfs.img takes nearly 30 minutes… If you do so, the low level format command

must be done before flashing all other files in including the parameter, boot, and rootfs files.

After flashing is completed, reboot the board. You'll notice the rootfs is still 1GB. Simply resize the EXT-4 partition:

```
resize2fs /dev/block/mtd/by-name/linuxroot
```

Let's check:

```
root@linaro-server:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mtdblock1  7.3G  546M  6.4G   8% /
udev            924M   84K  924M   1% /dev
none            4.0K     0  4.0K   0% /sys/fs/cgroup
none            186M  260K  185M   1% /run
none            5.0M     0  5.0M   0% /run/lock
none            927M     0  927M   0% /run/shm
none            100M     0  100M   0% /run/user
```

Yes! Success! The rootfs partition is 7.3 GB with 6.4GB available.

Installing LXDE or XFCE Desktop Environment

If you just plan to use Radxa Rock as a server, you are done, but if you want to run a desktop environment one more step is needed. Since there are no GPU Linux drivers in RK3188 based devices, installing Unity is probably a very bad idea and would lead to poor performance. Two other lightweight desktop environments are frequently used: LXDE in distributions such as Lubuntu or ALIP, or XFCE in distributions such as Xubuntu.

The good news is that installation is very easy, but it can take a few hours to download and install all required packages. So if you want to skip the part below, you can download the compressed rootfs with Xubuntu-Desktop to flash directly to your board.

Lubuntu (LXDE) requires 378 MB of archives and 1166 MB of additional disk space, and can be installed with the following command:

```
sudo apt-get install lubuntu-desktop
```

Xubuntu (XFCE) requires 452 MB of archives and 1370 MB of additional disk space, and can be installed with the following command :

```
sudo apt-get install xubuntu-desktop
```

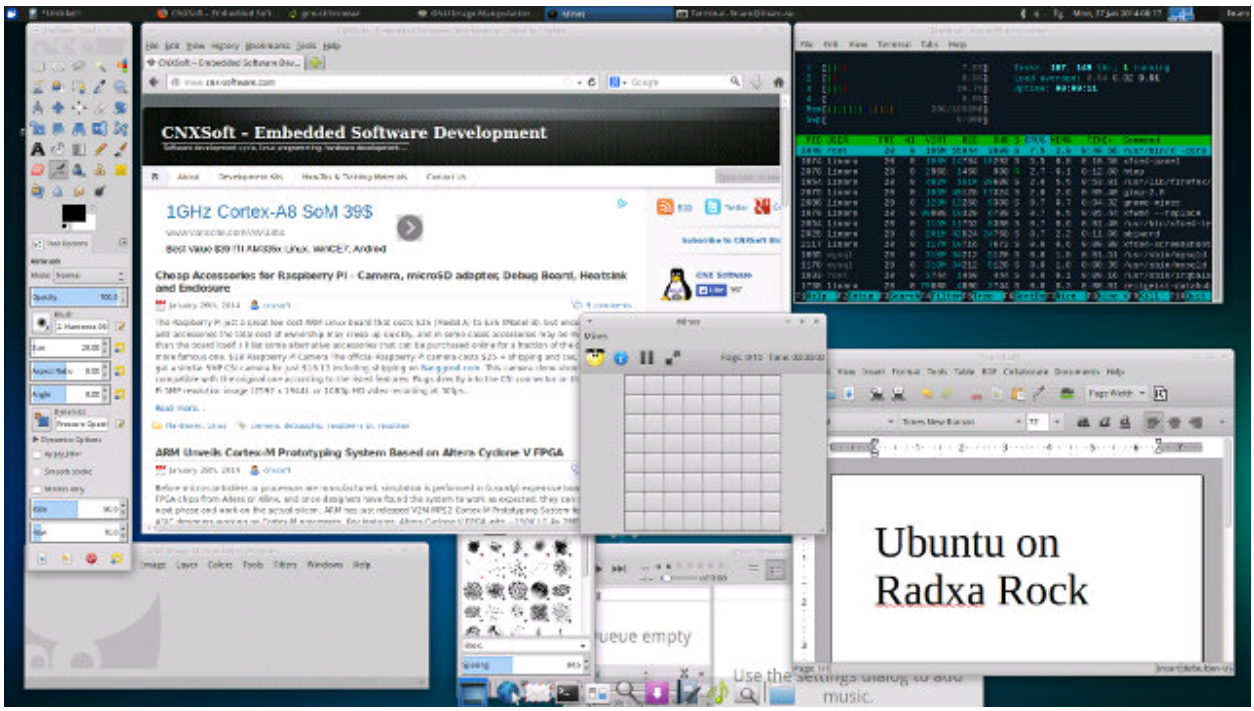
I decided to install Xubuntu desktop, and it took 2 to 3 hours. I'll upload my Ubuntu Saucy XFCE rootfs soon if you want to skip that part.

Once installation is complete we still have 4.5GB free, but somehow it used nearly 2GB additional space, instead of the 1370 MB promised.

```
df -h
```

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/mtdblock1  7.3G  2.5G  4.5G  36% /
udev            924M   84K  924M   1% /dev
none            4.0K     0  4.0K   0% /sys/fs/cgroup
none            186M  296K  185M   1% /run
none            5.0M     0  5.0M   0% /run/lock
none            927M     0  927M   0% /run/shm
none            100M     0  100M   0% /run/user
```

At boot time, you get the the login screen (use “linaro” user with “linaro” password), and the system is very usable. I’ve opened several programs including Gimp, Firefox, Abiword, a terminal window, a game, a music player, and more (See screenshot below), and everything runs pretty smoothly. Stability may need to be improved however, since I’ve got one random reboot as I accessed SAMBA shares. I could not reproduce the issue.



Xubuntu in Radxa Rock (Click for Original Size)

Related posts:

1. **Cross-compiling the ARM Linux Kernel in Ubuntu 12.04 LTS**
2. **Hardware Packs for AllWinner A10 Devices and Easier Method to Create a Bootable Ubuntu 12.04 SD Card**
3. **How-to Install and "Hack" Linux on T428 or MK802 IV mini PCs**
4. **Getting Started with Raxda Rock – Building an Android 4.2 Image from Source**
5. **Getting Started with MultiArch (armel / armhf) in Ubuntu**

Categories: Linux, Linux 3.0, Rockchip RK31xx, Ubuntu Tags: Linux, cross-compilation, how-to, kernel, linaro, lubuntu, lxde, radxa, tutorial, ubuntu, xfce, xubuntu
Comments (33) Trackbacks (3) Leave a comment Trackback

1. dr. Bormental
January 28th, 2014 at 03:09 | #1
Reply | Quote

screenshot from youtube (w/statictic activated would be good).

2. dr. Bormental
January 28th, 2014 at 03:13 | #2
Reply | Quote

> **dr. Bormental :**
> statictic

->statistics

3. Murat
January 28th, 2014 at 14:11 | #3
Reply | Quote

Hope one day they'll have gpu acceleration that also works in other rk3188 sticks.

4. cnxsoft
January 28th, 2014 at 21:58 | #4
Reply | Quote

@dr. Bormental
YouTube is not really usable. It can play some videos, but it's just too slow. I think I need to try the Chromebook flash library with Chromium:
http://hwswbits.blogspot.com/2013/04/flash-on-picuntu-linux-arm.html

5. 
cnxsoft
January 29th, 2014 at 11:51 | #5
Reply | Quote

I've uploaded the binaries, compressed with 7pzip:
boot.img – http://ubuntuone.com/4vlyMy53bKmi4pQsfb8r3C
Ubuntu server for Radxa Rock – http://ubuntuone.com/7UzDx4Wf9xlSCU8oNObIxI
Xubuntu for Radxa Rock – http://ubuntuone.com/2rM59DV6UE7QqcfHcMifNN

6. 
Toby Wintermute
January 30th, 2014 at 07:34 | #6
Reply | Quote

@cnxsoft
The Ubuntu server link is coming back as "Page Not Found" now.

7. 
cnxsoft
January 30th, 2014 at 09:28 | #7
Reply | Quote

@Toby Wintermute
Sorry, the link was wrong. I've updated it.

8. 
Alex
February 2nd, 2014 at 20:36 | #8
Reply | Quote

Great work, thank you! This is the first source to read of the command to flash the parameter file under Linux.

However, I did the whole process and it left me with a command line and no network connection although I configured the wired network interface correctly. The Radxa Rock system could be pinged from outside within my home network using its IP address, but could not ping out (error message "Socket – access denied" even as root) and could not resolve a network address (host google.com — timeout; could not the apt addresses for apt-get update). Looks like a tiny permission problem but I cannot figure it out.

Additionally, I cannot unpack your xz-files. Debian Linux xz from xz-utils cannot

recognize the file format, neither can xarchiver. Are they really xz-files (http://en.wikipedia.org/wiki/Xz) ?

Keep on the good work!

Alex

9. 
cnxsoft
February 2nd, 2014 at 20:49 | #9
Reply | Quote

@Alex
I compressed it with 7z.. I thought xz was also lzma, and I have used 7z to uncompressed xz files before… I should probably have used the extension 7z instead.

I had the same network problem as you have, using the default kernel kernel, and disabling CONFIG_ANDROID_PARANOID (Only allow certain groups to create sockets) fixed the issue. Are you using my boot.img?

10. 
Alex
February 2nd, 2014 at 23:55 | #10
Reply | Quote

@cnxsoft
Thanks a lot for the quick reply. It must be deep in the night in Thailand.
As I said, I could not open your packages so I used a self-compiled kernel.
I am not sure about CONFIG_ANDROID_PARANOID, but I thought I did that. But then maybe I used a different boot.img from an earlier attempt or an original from radxa.

I will try it again!

11. 
Marty
February 10th, 2014 at 05:41 | #11
Reply | Quote

Can boot.img and rock_rootfs.img be flashed to the RK3188 using Windows?

12. 
cnxsoft
February 10th, 2014 at 11:18 | #12
Reply | Quote

@Marty

Yes sure. Use RkAndroidTool. Don't forget to flash the parameter file too.

13. 
Marty
February 11th, 2014 at 04:12 | #13
Reply | Quote

@cnxsoft
You have the Linux parameter file here:

http://dl.radxa.com/rock/images/ubuntu/2013-12-17/

will this change when I use my own compiled boot.img and rock_roofs.img for Linux?

14. 
Cristian
February 14th, 2014 at 03:05 | #14
Reply | Quote

Hello, Great Post!! I make my firsts testing with Linaro on ARM, I have a question, if posible (via Windows) to make a copy to .IMG file of the running image ?? That is because I use 4 or 5 for office basic use, and change various settings as language, wifi settings, etc.
Thanks, Cristian.
Argentina.

15. 
Gilles
February 14th, 2014 at 08:00 | #15
Reply | Quote

Is it possible to install the images on a micro SD card (with dd) and boot from that card without altering the onboard NAND memory? Just like on the Cubieboard.
Thanks.

16. 
cnxsoft
February 14th, 2014 at 09:55 | #16
Reply | Quote

@Cristian
Yes, you can use RkFlashKit to backup the rootfs partition after you changed the settings, installed applications…

@Gilles
Yes. it should be possible, but I haven't tried it. You should be able to modify the parameter to mount the rootfs from the SD card, and copy the rootfs into your SD

card.

Follow the thread of google group, as details will hopefully emerge there:
https://groups.google.com/forum/#!msg/radxa/RhYUMGAxKTA/ZHBB80pNqYMJ

17. 
Marty
February 14th, 2014 at 23:51 | #17
Reply | Quote

I've had past success flashing the CX-919 in Windows using RkAndroidTool. So with that I need to flash my newly built boot.img and rock_rootfs.img along with the standard parameter file from above. What are the Adresses we should be using?

I used this setup, but it loads slow:

Address Name Path

0×00000000 parameter parameter
0×00008000 boot boot.img
0×00382000 linuxroot rock_rootfs.img

18. 
cnxsoft
February 15th, 2014 at 09:56 | #18
Reply | Quote

@Marty
I've also noticed it takes a minute to load. But I doubt the addresses in the parameter file have anything to do with the relatively slow boot time.

19. 
thehid
February 15th, 2014 at 17:56 | #19
Reply | Quote

Hey cnxsoft, I wrote to you on an other articel. You advised to use the Minix X7 SDK for building a wheezy or ubuntu image. I just downloaded the SDK and extracted the files, bit they looks like the other kernels for rk3188 on github. Would you give me a some more hints, how to build the kernel and boot.img. The rest seems to be manageable with this articel. Many thanks for your help!

20. 
cnxsoft
February 15th, 2014 at 18:40 | #20
Reply | Quote

@thehid

It should also be the same as above, except you need to use the defconfig for your product. The initrd part may also be different, and you could ask on Minix forums for that part if you can't make it work.

21. 
Marty
February 16th, 2014 at 02:57 | #21
Reply | Quote

@cnxsoft
Are the addresses I'm using the right ones? Maybe rootfs should go in System which is 0×00282000??

22. 
cnxsoft
February 16th, 2014 at 09:46 | #22
Reply | Quote

@Marty
You can actually choose any address you want. The respective partitions must just be big enough to store the parameter file, boot.img and the rootfs

23. 
Cristian
February 17th, 2014 at 12:41 | #23
Reply | Quote

Hello How are you…

Can any help me to configure RKAndroidTool to flash my own kernel, I having problems to flash (with RKAndroidTool v1.37) my built boot.img (12.960 MB) and rock_rootfs.img (1.048.576 MB) along with the parameter file:

CMDLINE:console=ttyFIQ0,115200 console=tty0 root=/dev/block/mtd/by-name/linuxroot rw rootfstype=ext4 init=/sbin/init initrd=0×62000000,0×00800000 mtdparts=rk29xxnand:0×00008000@0×00002000(boot),-@0x0000A000(linuxroot)

Address Name Path
0×00000000 parameter parameter
0×00008000 boot boot.img
0×00382000 linuxroot rock_rootfs.img

This are error on RKToolkit LOG are:
————————————-
02:27:55 145 INFO:Start to erase IDB
02:27:55 146 INFO:Start to lower format
02:28:06 175 INFO:Lower format OK
02:28:06 177 INFO:Erasing IDB OK

02:28:09 960 Image Thread Path(\\?
\USB#VID_2207&PID_310B#6&380B6F11&0&1#{a5dcbf10-6530-11d2-901f-00c04fb951ed}),ThreadID=6184
02:28:09 961 INFO:Start to test device
02:28:09 961 INFO:Start to run
02:28:17 974 INFO:Test device OK
02:28:17 975 INFO:Start to download file
02:28:17 982 INFO: Start to download files to flash,backup offset=0xffffffff
02:28:17 988 INFO:Flash Start to download file to flash
02:28:18 086 Info:RKA_File_Download start,file=boot,offset=0×8000
02:28:22 969 Info:RKA_File_Download start,file=linuxroot,offset=0×12000
02:29:28 799 INFO: Download files to flash finished,Total(281709960)B
02:29:28 799 INFO: Start to check data
02:29:28 800 INFO:Start to check download data
02:29:28 815 Error:Compare parameter file failed,Offset(0)
02:29:28 824 Error: Check parameter file failed
02:29:28 824 ERROR:Download file failed
02:29:28 825 ERROR:Image Thread Failed
02:29:28 827 ERROR:Run failed
————————————

With Server or Desktop boot-linux.img and rootfs-desktop.img or rootfs-server.img downloaded from RADXA site are the same problem. But, If try with “Batch Tool” and radxa images with “-update” Flash are OK.

Is very apreciated If any can send how are correct param file and RkAndroidTool settings to use for Flash custom Images/RootFs.

Thanks in Advance, Cristian.
Argentina.

24. 

cnxsoft
February 17th, 2014 at 12:58 | #24
Reply | Quote

@Cristian
Problem is with the parameter file. Try to use the one in that post. Or replace the dash before @0x0000A000(linuxroot) with a proper address.

25. 

Ze Sousa
February 19th, 2014 at 04:00 | #25
Reply | Quote

Hello. Please help.

I bought the rock and I want it to be my Desktop Linux computer, but always worked on Windows so don’t understand much of this.

I followed the section “Increasing Radxa Rock Ubuntu Rootfs Partition Size” because 1GB is not enough for a decent Operating system and when started the rock, it gave me the following errors in startup command line after a while:

mount: mounting /dev/block/mtd/by-name/linuxroot on /root failed: Invalid argument
(… some more mount errors…)
(initramfs)

Googling it for linux I only find comments like “put the CD wathever and run these commands”. But I’m in the rock, no CD drive.
What shall I do? Please make any answer easy to understand for newbies. Thanks in advance.

26. 
cnxsoft
February 19th, 2014 at 10:19 | #26
Reply | Quote

@Ze Sousa
How many partition are shown in RKFlashKit?

27. 
cnxsoft
February 19th, 2014 at 10:31 | #27
Reply | Quote

If you have problems with SAMBA share, somebody mentioned adding

socket options = TCP_NODELAY SO_SNDBUF=65535 SO_RCVBUF=65535

to the samba config (/etc/samba/smb.conf) appears to fix the issue.

28. 
Ze Sousa
February 20th, 2014 at 03:31 | #28
Reply | Quote

@cnxsoft
Hi,

I made it from Windows using RKAndroidTool.
I find that RKFlashKit as a Linux app, I don’t have a Linux machine.

29. 
Ze Sousa
February 20th, 2014 at 05:06 | #29
Reply | Quote

> **Ze Sousa :**
> @cnxsoft
> Hi,
> I made it from Windows using RKAndroidTool.
> I find that RKFlashKit as a Linux app, I don't have a Linux machine.

More specifically:

** start
1. open RKAndroidTool and connect the rock
2. erase IDB
3. run parameter file changed acoordingly to this tutorial
4. run files linuxroot and boot (only)
** end

Is anything missing here?

30. 
cnxsoft
February 20th, 2014 at 11:18 | #30
Reply | Quote

@Ze Sousa
It looks correct. Any error messages in RkAndroidTool?

31. 
Ze Sousa
February 20th, 2014 at 19:03 | #31
Reply | Quote

@cnxsoft
Yes, in the verification after run linuxroot and boot.
But ran it again and everything gone OK.
I'll do it again step by step and write down any errors or warnings here.

32. 
Ze Sousa
February 21st, 2014 at 03:33 | #32
Reply | Quote

> **Ze Sousa :**
> @cnxsoft
> Yes, in the verification after run linuxroot and boot.
> But ran it again and everything gone OK.
> I'll do it again step by step and write down any errors or warnings here.

This time everything went OK.
Sorry for everything, thank you for your time.

33. 

cnxsoft
March 16th, 2014 at 10:55 | #33
Reply | Quote

Updated images for Android, Linux, and Dual boot, as now available @ http://radxa.com/download/

ChangeLog:

Android

Remove the linuxroot partitions for pure Android image, this gives 1.5G more space for /sdcard in Android
Support the 13 key simple remote from radxa,
Support 720p HD USB camera, you can now skype on RR
Support Bluetooth keyboard/mouse
Support USB OTG host mode, mounting USB disk/HDD via OTG
Wifi driver stability and improvement
Enable MTP and adb mode at the same time

Linux

Expand the rootfs to the whole NAND(8G for full version, 4G for lite version)
Add kernel cmd line to pass mac address for a fixed ip address
Enable display scale for TVs which has problems with overscan, see display for how to scale the screen
Support wifi onRR full version(RTL8723au) and lite version(RTL8188ETV) read here to configure it
Fix rebooting when Ethernet uploading problem see discussion here
Fix mysql/apache permission issue
Fix chromium not rendering web page issue
Add support for Flash player in chromium

1. May 18th, 2014 at 14:20 | #1
   Ubuntu 14.04 and Slitaz on Radxa Rock
2. May 25th, 2014 at 10:54 | #2
   How to Upgrade Firmware for Rockchip RK3066/RK3188 Devices with the Command Line in Linux
3. July 2nd, 2014 at 21:20 | #3
   Easy and Safe Way to Try Linux on Popular Rockchip RK3188 mini PCs

	Name (required)
	E-Mail (will not be published) (required)
	Website

Subscribe to comments feed

Submit Comment

How to Install XBMC on D-Link Boxee Box Cheap Accessories for Raspberry Pi – Camera, microSD adapter, Debug Board, Heatsink and Enclosure

RSS

Twitter

Google +

Subscribe to CNXSoft Blog by Email

CNX Software

Like 1,195

Recent Comments

onebir on Easy and Safe Way to Try Linux on Popular Rockchip RK3188 mini PCs

Gabe on List of RK3188 mini PCs / HDMI TV Dongles

cnxsoft on Hummingbird A31 Is a $70 Feature-packed Development Board Powered by AllWinner A31

gicho on Forlinx OK335xS-II is a Low Cost Linux and Wince Industrial Board Powered by TI Sitara AM335x Processor

linuxium on Easy and Safe Way to Try Linux on Popular Rockchip RK3188 mini PCs

Graham on Hummingbird A31 Is a $70 Feature-packed Development Board Powered by AllWinner A31

Sponsors

Buy Development Kits, Tablets and STB

Tags

Android antutu arduino arm benchmark big little bsp debian development board development kit devkit digital signage freescale gpu how-to ics intel IoT jelly bean kernel kickstarter kitkat linaro Linux mcu media player mini-pc open source pandaboard qualcomm raspberry pi review samsung sdk set-top box smartphone som stb tablet texas instruments tutorial ubuntu windows embedded xbmc yocto

Blogroll

- Free Embedded Development E-Books
- Investment Ideas
- MHL TV
- Thai Translation

Top WordPress