

Linux

How to download and compile the Linux system kernel for **ODROID-XU3/4**. Generic Linux system needs **gcc version 4.9** to build the Kernel.

Step-by-Step guide to building an ODROID-XU3/4 Kernel

Download the cross tool chain package

[<http://dn.odroid.com/ODROID-XU/compiler/arm-eabi-4.6.tar.gz>]

Note

- This toolchain is only used to build the kernel.
- This toolchain is included in the Android source package.
(`$ANDROID_ROOT/prebuilts/gcc/linux-x86/arm/arm-eabi-4.6`)

Copy the cross tool package to /opt/toolchains

If the '/opt/toolchains' directory does not exist in host pc, then create the directory.

```
$ sudo mkdir /opt/toolchains
$ sudo cp arm-eabi-4.6.tar.gz /opt/toolchains
```

Uncompress the cross tool with tar command

```
$ cd /opt/toolchains
$ sudo tar xvfz arm-eabi-4.6.tar.gz
```

Add Path in your environment file

Modify your `~/.bashrc` file to add a new path with editor (gedit or vi)

```
export ARCH=arm
export PATH=${PATH}:/opt/toolchains/arm-eabi-4.6/bin
export CROSS_COMPILE=arm-eabi-
```

To apply this change, login again or restart the `.bashrc`

```
$ source ~/.bashrc
```

Check the tool-chain path to check it is set up correctly.

```
$ arm-eabi-gcc -v
Using built-in specs.
COLLECT_GCC=arm-eabi-gcc
COLLECT_LTO_WRAPPER=/opt/toolchain/arm-eabi-4.6/bin/../libexec/gcc/arm-
eabi/4.6.x-google/lto-wrapper
Target: arm-eabi
Configured with: /tmp/android-15472/src/build/./gcc/gcc-4.6/configure --
prefix=/usr/local --target=arm-eabi --host=x86_64-linux-gnu
--build=x86_64-linux-gnu --with-gnu-as --with-gnu-ld --enable-
languages=c,c++ --with-gmp=/tmp/android-15472/obj/temp-install --with-
mpfr=/tmp/android-15472/obj/temp-install --with-
mpc=/tmp/android-15472/obj/temp-install --without-ppl --without-clog --
disable-libs
sp --enable-threads --disable-nls --disable-libmudflap --disable-libgomp --
disable-libstdc__-v3 --disable-sjlj-exceptions --disable-
shared --disable-tls --disable-libitm --with-float=soft --with-fpu=vfp --
with-arch=armv5te --enable-target-optspace --with-abi=aapcs
--with-gcc-version=4.6 --with-binutils-version=2.21 --with-gmp-
version=4.2.4 --with-mpfr-version=2.4.1 --with-gdb-version=7.3.x --w
ith-arch=armv5te --with-sysroot=/tmp/android-15472/install/sysroot --with-
prefix=/tmp/android-15472/install --with-gold-version=2.21
--enable-gold --disable-gold --disable-multilib --program-transform-
name='s&^&arm-eabi-&'
Thread model: single
gcc version 4.6.x-google 20120106 (prerelease) (GCC)
```

Linux

Linux ODROID-XU3/4 works as follow:

It must have at least two partitions. First Partition must be a FAT32/EXT4 partition. Second Partition can be whatever the Filesystem that your kernel supports (must be built in).

Note: It is possible to use the first partition as ext4, however it is strongly not recommended due to Windows Users losing the capability of changing boot.ini

Partition Contents

Partition 1:

- Kernel Image (zImage)

- boot.scr
- exynos5422-odroidxu3.dtb
- ulnitrd (if applicable)

Partition 2:

- rootfs (a.k.a. File System)

Currently Supported Linux Distributions

- Ubuntu 16.04 [Software Release for Linux/Ubuntu on XU4/XU3](#)

Note: More distribution support will come with time.

HDMI Support On Linux

HDMI support should work out-of-box for everyone including framebuffer console. If you experience any issue, please contact us on [ODROID Forums](#)

On the provided Ubuntu image there are several examples on how to configure the HDMI to your specific resolution or even lock to a certain resolution. You can check the configuration file on `/media/boot/boot.ini` of the Ubuntu Image

DisplayPort Support on Linux

For displayport configuration, please follow this guide [Displayport Guide on ODROID Forums](#)

Kernel Sources

Kernel sources for ODROID-XU3/4 is on our [Github](#). Branch are **odroidxu3-3.10.y** or **odroidxu4-4.9.y** defconfig are **odroidxu3_defconfig** or **odroidxu4_defconfig**

Kernel Rebuild Guide

Note

You must complete these steps on the **ODROID-XU3/4 board**. You can compile kernel & dtb images on the host PC using the cross compiler but you cannot create ulnitrd ramdisk image on the host PC since the binary files in ramdisk image. Refer to current root file system during the executing 'update-initramfs' command.

Please follow the instructions below to rebuild the Linux Kernel for ODROID. Those instructions cover native build of the Kernel.

3.10.y & 4.9.y

Install dependencies:

```
odroid@odroid:~$ sudo apt-get install build-essential libqt4-dev  
libcurses5-dev git
```

Clone Repo:

odroidxu3-3.10.y

```
odroid@odroid:~$ git clone --depth 1 https://github.com/hardkernel/linux.git  
-b odroidxu3-3.10.y odroidxu3-3.10.y  
odroid@odroid:~$ cd odroidxu3-3.10.y
```

odroidxu4-4.9.y

```
odroid@odroid:~$ git clone --depth 1 https://github.com/hardkernel/linux.git  
-b odroidxu4-4.9.y odroidxu4-4.9.y  
odroid@odroid:~$ cd odroidxu4-4.9.y
```

Configure Kernel:

```
odroid@odroid:~$ make odroidxu3_defconfig
```

Do changes if you need/want:

```
odroid@odroid:~$ make menuconfig
```

Build Kernel and Modules:

```
odroid@odroid:~$ make -j8
```

**This explanation assumes that your USB memory CARD reader is assigned at /dev/sdb.
Be careful!**

Install zImage & DTB file:

```
odroid@odroid:~$ mkdir -p mount  
odroid@odroid:~$ sudo mount /dev/sdb1 ./mount
```

```
odroid@odroid:~$ sudo cp arch/arm/boot/zImage arch/arm/boot/dts/exynos5422-odroidxu3.dtb /media/boot && sync && sudo umount ./mount
```

Install Modules:

```
odroid@odroid:~$ sudo mount /dev/sdb2 ./mount
odroid@odroid:~$ sudo make modules_install ARCH=arm INSTALL_MOD_PATH=./mount
&& sync && sudo umount ./mount
odroid@odroid:~$ rm -rf mount
```

Copy .config to /boot for initramfs creation:

```
odroid@odroid:~$ sudo cp .config /boot/config-`make kernelrelease`
```

Create initramfs:

```
odroid@odroid:~$ sudo update-initramfs -c -k `make kernelrelease`
```

Create ulnitrd:

```
odroid@odroid:~$ sudo mkimage -A arm -O linux -T ramdisk -C none -a -e -n uInitrd -d /boot/initrd.img-`make kernelrelease` /boot/uInitrd-`make kernelrelease`
```

Install new ulnitrd:

```
odroid@odroid:~$ sudo cp /boot/uInitrd-`make kernelrelease` /media/boot/uInitrd
```

reboot:

```
odroid@odroid:~$ sudo sync && reboot
```

Your new kernel should be installed. Note for your own convenience we provide daily builds of Linux kernel that can be easily installed on Supported distros by using a ODROID Utility.

```
odroid@odroid:~$ sudo -s
root@odroid:~$ wget -O /usr/local/bin/odroid-utility.sh
https://raw.githubusercontent.com/mdrjr/odroid-utility/master/odroid-utility.sh
root@odroid:~$ chmod +x /usr/local/bin/odroid-utility.sh
root@odroid:~$ odroid-utility.sh
```

4.14.y

Native-build

This guide is only for the **NATIVE BUILD**.
run it on the board.

Installing building tools

You have to install some building tools and related packages.

```
$ sudo apt install git gcc g++ build-essential libssl-dev bc
```

Download and build the kernel source.

Updating Kernel and DTB (Device Tree Blob)

Please note that native kernel compile on ODROID-XU4 will take about 25 minutes.

```
$ git clone --depth 1 https://github.com/hardkernel/linux -b  
odroidxu4-4.14.y  
$ cd linux  
$ make odroidxu4_defconfig  
$ make -j8  
$ sudo make modules_install  
$ sudo cp -f arch/arm/boot/zImage /media/boot  
$ sudo cp -f arch/arm/boot/dts/exynos5422-odroidxu3.dtb /media/boot  
$ sudo cp -f arch/arm/boot/dts/exynos5422-odroidxu4.dtb /media/boot  
$ sudo cp -f arch/arm/boot/dts/exynos5422-odroidxu3-lite.dtb /media/boot  
$ sync
```

Updating root ramdisk (Optional)

```
$ sudo cp .config /boot/config-`make kernelrelease`  
$ sudo update-initramfs -c -k `make kernelrelease`  
$ sudo mkimage -A arm -O linux -T ramdisk -C none -a -e -n uInitrd -d  
/boot/initrd.img-`make kernelrelease` /boot/uInitrd-`make kernelrelease`  
$ sudo cp /boot/uInitrd-`make kernelrelease` /media/boot/uInitrd  
$ sync
```

Before you start with new Linux kernel v4.14

You would check all necessary files are in place as below before reboot. The file size would differ.

```
$ ls -l /media/boot/  
total 14756
```

```
-rwxr-xr-x 1 root root 9536 Oct 25 23:29 boot.ini
-rwxr-xr-x 1 root root 753 Aug 20 22:38 boot.ini.default
-rwxr-xr-x 1 root root 62565 Nov 2 01:24 exynos5422-odroidxu3.dtb
-rwxr-xr-x 1 root root 61814 Nov 2 01:24 exynos5422-odroidxu3-lite.dtb
-rwxr-xr-x 1 root root 62225 Nov 2 01:24 exynos5422-odroidxu4.dtb
-rwxr-xr-x 1 root root 61714 Oct 25 23:30 exynos5422-odroidxu4-kvm.dtb
-rwxr-xr-x 1 root root 9996513 Nov 2 01:27 uInitrd
-rwxr-xr-x 1 root root 4844744 Nov 2 01:24 zImage
```

```
$ sudo sync
$ sudo reboot
```

Cross-build

This guide is only for the **CROSS BUILD**.
run it on the Ubuntu desktop.

Installing required packages

You will need install required packages before you start to build Linux kernel on your Ubuntu desktop.
You will need install required packages before you start to build Linux kernel on your Ubuntu desktop.

```
$ sudo apt update
$ sudo apt install git
$ sudo apt install lib32stdc++6 lib32z1 lzop u-boot-tools
$ sudo apt install build-essential gcc
$ sudo apt install libncurses5-dev
```

Download the cross toolchain

Download

```
$ sudo mkdir -p /opt/toolchains
$ sudo tar xvf gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz -C /opt/toolchains/
```

In order to add the toolchain path to PATH, paste below lines to **\$HOME/.bashrc**.

```
export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabi-
export PATH=/opt/toolchains/gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi/bin/:$PATH
```

Download and Build Kernel

```
$ git clone https://github.com/hardkernel/linux -b odroidxu4-4.14.y
```

```
$ cd linux
$ make odroidxu4_defconfig
$ make -j8
```

Updating Kernel and DTB (Device Tree Blob)

This explanation assume that your USB memory CARD reader is assigned at /dev/sdc. Be careful!

A partition for kernel, dtb and initrd is necessary. Run `fdisk <device>` and create a **root partition** (partition 1).

If you plan to have your rootfs on the sdcard on another partition you can put a minimum of +60M for the last sector of this root partition.

Change type to FAT16 ("t" option, value 6) and type "w" to save changes. Format it with `mkfs.vfat /dev/sd<device>1` and put your kernel, dtb, boot.ini and initrd there.

1. Plug the Boot-Device(eMMC or SD) into the USB memory CARD reader and Connect the USB memory CARD reader to your HOST PC(Linux OS).
2. Copy the zImage and DT((exynos5422-odroidxu?.dtb) to the FAT partition(1st partition) in the Boot-Device.

```
$ mkdir -p mount
$ sudo mount /dev/sdc1 ./mount
$ sudo cp arch/arm/boot/zImage arch/arm/boot/dts/meson8b_odroidxu?.dtb
./mount && sync && sudo umount ./mount
```

3. Copy the driver modules to the EXT4 partition(2nd partition) in the Boot-Device.

```
$ sudo mount /dev/sdc2 ./mount
$ sudo make modules_install ARCH=arm INSTALL_MOD_PATH=./mount && sync &&
sudo umount ./mount
$ rm -rf mount
```


From:

<http://wiki.odroid.com/> - **ODROID Wiki**

Permanent link:

http://wiki.odroid.com/odroid-xu4/software/building_kernel

Last update: **2018/11/18 01:46**

