

Linux

This page introduce how you can download and compile the Linux system kernel for **ODROID-U3**. We recommend to use the **gcc version 4.7.3** for Kernel building.

Toolchain

Click one of the site to download toolchain to build Linux system kernel.

- [Download #1](#)
- [Download #2](#)

Once the download is done, extract the tarball to **/opt/toolchains/**.

```
$ sudo mkdir -p /opt/toolchains
$ sudo tar jxvf gcc-linaro-arm-linux-
gnueabi-4.7-2012.12-20121214_linux.tar.bz2 -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-arm-linux-
gnueabi-4.7-2013.04-20130415_linux/bin:$PATH
```

You can apply the change if you login again or import to apply this change, login again or evaluate **\$HOME/.bashrc** with source command.

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

You can check if the toolchain installed above works properly while checking the version of toolchain. If you can find **gcc version 4.7.3 20130328 (prerelease)** at the end of the line, the toolchain is well installed.

```
$ arm-linux-gnueabi-gcc -v
Using built-in specs.
COLLECT_GCC=arm-linux-gnueabi-gcc
COLLECT_LTO_WRAPPER=/opt/toolchains/gcc-linaro-arm-linux-
gnueabi-4.7-2013.04-
20130415_linux/bin/./libexec/gcc/arm-linux-gnueabi/4.7.3/lto-wrapper
Target: arm-linux-gnueabi
Configured with: /cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-
gnueabi-linux/.build/src/gcc-linaro-4.7-
2013.04/configure --build=i686-build_pc-linux-gnu --host=i686-build_pc-
linux-gnu --target=arm-linux-gnueabi --
prefix=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-
linux/install --with-
```

```
sysroot=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/install/arm-linux-gnueabi-hf/libc --
enable-languages=c,c++,fortran --enable-multilib --with-arch=armv7-a --with-
tune=cortex-a9 --with-fpu=vfpv3-d16 --
with-float=hard --with-pkgversion='crosstool-NG
linaro-1.13.1-4.7-2013.04-20130415 - Linaro GCC 2013.04' --with-
bugurl=https://bugs.launchpad.net/gcc-linaro --enable-__cxa_atexit --enable-
libmudflap --enable-libgomp --enable-
libssp --with-gmp=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-
gnueabi-hf-linux/.build/arm-linux-
gnueabi-hf/build/static --with-mpfr=/cbuild/slaves/oorts/crosstool-
ng/builds/arm-linux-gnueabi-hf-linux/.build/arm-
linux-gnueabi-hf/build/static --with-mpc=/cbuild/slaves/oorts/crosstool-
ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static --with-
ppl=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-
gnueabi-hf-linux/.build/arm-linux-gnueabi-hf/build/static --with-
clog=/cbuild/slaves/oorts/crosstool-ng/builds/arm-
linux-gnueabi-hf-linux/.build/arm-linux-gnueabi-hf/build/static --with-
libelf=/cbuild/slaves/oorts/crosstool-
ng/builds/arm-linux-gnueabi-hf-linux/.build/arm-linux-gnueabi-hf/build/static
--with-host-libstdcxx=' -
L/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static/lib -
lpwl' --enable-threads=posix --disable-libstdcxx-pch --enable-linker-build-
id --enable-gold --with-local-
prefix=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/install/arm-linux-gnueabi-hf/libc --
enable-c99 --enable-long-long --with-mode=thumb
Thread model: posix
gcc version 4.7.3 20130328 (prerelease) (crosstool-NG
linaro-1.13.1-4.7-2013.04-20130415 - Linaro GCC 2013.04)
```

If your host PC has any x86-32bit library compatibility issue, install the related packages.

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386
```

Checkout

You can checkout **Kernel** source tree from [Hardkernel's Github](https://github.com/hardkernel/linux). Please note that we distribute the Linux kernel in different branches for Android and other Linux distributions.

```
$ git clone --depth 1 https://github.com/hardkernel/linux.git -b
odroid-3.8.y
$ cd linux
```

Compile

Before compiling, you must configure for **ODROID-U3** with following command.

```
$ make odroidu_defconfig
$ make
```

Then you can start to build Linux kernel image, **zImage**. This will create **arch/arm/boot/zImage**. You can also add **-j** option for fast compile.

Installation

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

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 to the FAT partition(1st partition) in the Boot-Device.

```
$ mkdir -p mount
$ sudo mount /dev/sdc1 ./mount
$ sudo cp arch/arm/boot/zImage ./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/old_product/odroid-x_u_q/odroid_u3/u3_building_kernel

Last update: **2017/08/01 09:07**

