

U-Boot (v2017.05)

There are two branches of XU3/4 u-boot in the Hardkernel github.
Since 25 Aug, 2017, **odroidxu4-v2017.05 based on u-boot mainline v2017.05** is available and odroidxu4-v2017.05 will be maintained as an official version of XU3/4 Ubuntu and Android.

If you need to use u-boot v2012.07, please refer to the following link.
[U-Boot v2012.07](#)

Toolchain

Click one of the site to download toolchain to build U-boot. Please note that this toolchain is for U-boot, there is another toolchain to build Linux kernel.

- [Download Bootloader Tool chain\(arm-linux-gnueabi-hf-4.9\)](#)

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

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

```
$ sudo mkdir -p /opt/toolchains
$ sudo unxz gcc-linaro-arm-linux-gnueabi-hf-4.9-2014.09_linux.tar.xz
$ sudo tar xf gcc-linaro-arm-linux-gnueabi-hf-4.9-2014.09_linux.tar -C
/opt/toolchains/
```

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

```
export PATH=/opt/toolchains/gcc-linaro-arm-linux-
gnueabi-hf-4.9-2014.09_linux/bin/:$PATH
export CROSS_COMPILE=arm-linux-gnueabi-hf-
export ARCH=arm
```

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 4.9.2** at the end of the line, the toolchain is well installed.

```
$ arm-linux-gnueabi-hf-gcc -v
Using built-in specs.
```

```
COLLECT_GCC=arm-linux-gnueabi-gcc
COLLECT_LTO_WRAPPER=/opt/toolchains/gcc-linaro-arm-linux-gnueabi-4.9-2014.09_linux/bin/./libexec/gcc/arm-linux-gnueabi/4.9.2/lto-wrapper
Target: arm-linux-gnueabi
Configured with: /cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/src/gcc-linaro-4.9-2014.09/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-linux/install/arm-linux-gnueabi/libc --enable-languages=c,c++,fortran --disable-multilib --enable-multiarch --with-arch=armv7-a --with-tune=cortex-a9 --with-fpu=vfpv3-d16 --with-float=hard --with-pkgversion='crosstool-NG linaro-1.13.1-4.9-2014.09 - Linaro GCC 4.9-2014.09' --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-linux/.build/arm-linux-gnueabi/build/static --with-mpfr=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-mpc=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-isl=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-cloog=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-libelf=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --enable-threads=posix --disable-libstdcxx-pch --enable-linker-build-id --enable-plugin --enable-gold --with-local-prefix=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/install/arm-linux-gnueabi/libc --enable-c99 --enable-long-long --with-mode=thumb --disable-multilib --with-float=hard
Thread model: posix
gcc version 4.9.2 20140904 (prerelease) (crosstool-NG linaro-1.13.1-4.9-2014.09 - Linaro GCC 4.9-2014.09)
```

Checkout & compile

You can checkout **U-boot** source tree from [Hardkernel's Github](#).

```
$ git clone https://github.com/hardkernel/u-boot.git -b odroidxu4-v2017.05
```

Before you compile **U-boot**, you must configure for **ODROID-XU3/4** with following command.

```
$ cd u-boot
$ make odroid-xu4_defconfig
```

And finally you can compile **U-boot**.

```
$ make
```

Then you can start to build u-boot image, **u-boot-dtb.bin**. This will create **u-boot/u-boot-dtb.bin**. You can also add **-j** option for fast compile.

Installation

ODROID-XU3/4 must have the secure boot enabled boot loader. There are four components of boot loader bl1.bin.hardkernel, bl2.bin.hardkernel.720k_uboot, u-boot-dtb.bin and tzsw.bin.hardkernel. u-boot-dtb.bin file can be generated by u-boot source code. But bl1.bin.hardkernel, bl2.bin.hardkernel.720k_uboot and tzsw.bin.hardkernel blobs can be found in u-boot/sd_fuse directory.

- **If you want to contribute your patch, apply it to our u-boot github admin. Or, contact “odroid.uboot@gmail.com”**
- To flash the updated images on your [MicroSD](#) card, refer the scriptor file `sd_fusing.sh` in `sd_fuse` directory.
- **To update the boot loader area in the eMMC, must use the [Fastboot Tool](#).**
- **If the eMMC boot loader area is corrupted, use the [eMMC Recovery](#) process.**
- For eMMC update, you need a fastboot driver and micro-USB cable.

Installation to blank card

We provide the script, **sd_fuse/sd_fusing.sh**, this helps you to install the bootloader into your blank card [MicroSD](#).

1. Insert your card to USB card reader and attach to USB host port of your desktop.
2. Check the device path of your USB card reader.
3. Install the bootloader binaries using **sd_fuse/sd_fusing.sh**.

```
$ cd sd_fuse
$ ./sd_fusing.sh <device/path/of/your/card>
```

Installation using fastboot

If you can boot your **ODROID-XU3/4** already and want to install a new **u-boot-dtb.bin** built by you. Fastboot helps you to install a **u-boot-dtb.bin** into your board.

u-boot-dtb.bin install

```
$ sudo fastboot flash bootloader u-boot-dtb.bin
```

bl1.bin install

```
$ sudo fastboot flash fwbl1 sd_fuse/bl1.bin.hardKernel
```

bl2.bin install

```
$ sudo fastboot flash bl2 sd_fuse/bl2.bin.hardKernel_720k_uboot
```

tzsw.bin install

```
$ sudo fastboot flash tzsw sd_fuse/tzsw.bin.hardKernel
```

If installation is done, you care reboot your **ODROID-XU3/4** with fastboot.

```
$ sudo fastboot reboot
```

Update the u-boot binary in SD card on Linux command line (or on the board)

You also can update U-boot from Linux with **dd** command. After building U-boot, copy **u-boot-dtb.bin** into your **ODROID-XU3/4**. Then do the command below in order to flash your U-boot image to Micro SD card.

```
$ sudo dd if=u-boot-dtb.bin of=<device/path/of/your/card> seek=63  
$ sudo sync
```

Update the u-boot binary in eMMC module on Linux command line

Write u-boot image to eMMC module on the XU4 board Linux command line.

1. Assume you already copied the compiled u-boot-dtb.bin file into the eMMC boot(FAT) partition
2. Boot from the eMMC module.
3. Open terminal and type "sudo su" to become a super user.
4. Unlock the eMMC boot blocks.

```
echo > /sys/block/mmcblk0boot0/force_ro
```

5. Copy the u-boot binary to the eMMC boot block.

```
dd if=/media/boot/u-boot-dtb.bin of=/dev/mmcblk0boot0 bs=512 seek=62
```

6. Flush and reboot.

```
sync && reboot
```

And you can see your own uboot build information from the USB-UART console terminal.

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

Permanent link:
http://wiki.odroid.com/odroid-xu4/software/building_u-boot_mainline

Last update: **2018/03/15 10:04**

