# Android

This page will introduce you how you can download and build Android source tree to customize it for your usage.

> If you have not built Android on your desktop yet, please read Android's official build environment set up guide https://source.android.com/source/initializing.html carefully before your proceed. Otherwise, you would have unexpected errors and exhaust to read quite long and complicated build log to find the problem. Also note that you need around 100GB of free space in your storage to get the full source code and binary space. At least 8GB of RAM is required. Android 5.x (Lollipop) build needs the OpenJDK 7 not the Oracle-Java-JDK. https://source.android.com/source/requirements.html

# Toolchain

For Linux kernel and U-boot, you must install specific toolchains. Android will use its prebuilt ARM toolchain.

**U-boot**

Toolchain

**Linux kernel**

Toolchain

# Checkout full source code

This command will initiate to download Android source tree for **ODROID-C1**.

**Android kitkat**

**4.4.2 v3.2**

```
$ mkdir odroid-c1
$ cd odroid-c1
$ repo init -u https://github.com/hardkernel/android.git -b
s805_4.4.2_master
$ repo sync
```

**4.4.4 v3.3**

```
$ mkdir odroid-c1
$ cd odroid-c1
$ repo init -u https://github.com/hardkernel/android.git -b
s805_4.4.4_master
$ repo sync
```

## Android lollipop

```
$ mkdir odroid-c1
$ cd odroid-c1
$ repo init -u https://github.com/hardkernel/android.git -b
s805_5.1.1_master
$ repo sync
```

# Build

Please make sure that you've already installed cross toolchain for kernel and U-boot before you start to build.

- Toolchain for U-boot
- Toolchain for Kernel

There is no specific build script for **ODROID-C1**, instead it supports generic build step what AOSP does as below.

## Android Kitkat

```
$ export ARCH=arm
$ export CROSS_COMPILE=arm-linux-gnueabihf-
$ export PATH=/opt/toolchains/gcc-linaro-arm-linux-
gnueabihf-4.9-2014.09_linux/bin/:$PATH
$ export PATH=/opt/toolchains/gcc-linaro-arm-none-
eabi-4.8-2014.04_linux/bin:$PATH
$ export JAVA_HOME=/usr/lib/jvm/java-6-oracle
```

```
$ export PATH=$JAVA_HOME/bin:$PATH
$ source build/envsetup.sh
$ lunch odroidc-eng
$ make -j4
```

## Android Lollipop

```
$ export ARCH=arm
$ export CROSS_COMPILE=arm-linux-gnueabihf-
$ export PATH=/opt/toolchains/gcc-linaro-arm-linux-
gnueabihf-4.9-2014.09_linux/bin/:$PATH
$ export PATH=/opt/toolchains/gcc-linaro-arm-none-
eabi-4.8-2014.04_linux/bin:$PATH
$ export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-amd64
$ export PATH=$JAVA_HOME/bin:$PATH
$ source build/envsetup.sh
$ lunch odroidc-eng
$ make -j4
```

You can replace the number on the command **make** for simultaneous compile as much as number of core on your build machine. But it should not to exceed 16, otherwise build would stop with an error.

# Download

Once Android build is completed, you will have all necessary files to download into your MicroSD/eMMC. If you already have a boot media and can switch to **fastboot** mode in U-boot, you can proceed to update while executing below commands one by one or ignore particular steps.
How to use fastboot

## Boot loader

```
$ fastboot flash bootloader out/target/product/odroidc/u-boot.bin
```

## Linux kernel

```
$ fastboot flash dtb out/target/product/odroidc/meson8b_odroidc.dtb
$ fastboot flash boot out/target/product/odroidc/kernel
```

## Android

These three files contains Linux utilities, Android frameworks and pre-installed Android applicatioins as well as necessary files to run Android. These three files will be flash into its own partition dedicated, please find the partition table pre-defined for each partition's size.

### KitKat

```
$ fastboot flash system out/target/product/odroidc/rootsystem.img
$ fastboot flash userdata out/target/product/odroidc/userdata.img
$ fastboot flash cache out/target/product/odroidc/cache.img
```

### Lollipop

```
$ fastboot flash system out/target/product/odroidc/rootsystem.img
$ fastboot flash cache out/target/product/odroidc/cache.img
```

## Misc

These file are not mandatory for Android to use, but useful for further features. **recovery.img** helps you to update Android system, **ODROID-C1** use this for installatioin. **hardkernel-720.bmp** is to show a logo on your HDMI display on boot and disappear when Android display subsystem is initiated.

### Kikkat

```
$ fastboot flash recovery out/target/product/odroidc/recovery.img
$ fastboot flash logo out/target/product/odroidc/hardkernel-720.bmp
```

### Lollipop

```
$ fastboot flash recovery out/target/product/odroidc/recovery.img
$ fastboot flash logo out/target/product/odroidc/hardkernel-720.bmp.gz
```

# How to create Android bootable media with empty MicroSD/eMMC

## Proceed each steps

### Using Self-Installation Image

This is the same way how Hardkernel team create Android release image and most easiest way to have bootable media on your **ODROID-C1** if you proceed one more build step. This procedure combine all required image files created by Android build into single file, **out/target/product/odroidc/selfinstall-odroidc.bin**.

```
$ make selfinstall
```

## Using Updatepackage.zip

**out/target/product/odroidc/updatepackage-odroidc-signed.zip**.

```
$ make updatepackage
```

Copy updatepackage-odroidc-signed.zip via USB mass storage and run ODROID Utility and Select "Package install from storage".

# FAQ

## How can I get into fastboot mode to download Android images

You need **USB-serial kit**. You must press any key on your keyboard when **ODROID-C1** is started. This will stop booting and give you **U-boot** command line, just execute the command **fastboot**.

## I do not have a boot media yet, how can I download Android images to my MicroSD or eMMC?

Obviously you must have a boot media can boot from **ODROID-C1** directly in order to enter **fastboot** mode in U-boot. Otherwise you are not able to download Android image to dedicated partitions (**system/userdata/cache**). The image files for such partitioins can not be downloaded from a desktop directly with **dd**, for example, since these images are compressed and **fastboot** extract while installing for each partitions.

## I do not have USB-serial kit, so I am not able to get into U-boot

The current **U-boot** does not support any other input device except serial console, hence you need USB-serial kit. Possible work around is to install a released Android image by Hardkernel team, and execute a command **reboot fastboot** from Android's shell command line. It will restart **ODROID-C1** and stop on U-boot to get into **fastboot** mode. You can execute command **fastboot** from your desktop while USB cable is connected with **ODROID-C1**.

## Why don't we download boot.img created by Android build

**ODROID-C1**'s Android file system is little bit different with others for easy customizing. We made another Android system image file, **rootsystem.img**, which is combined with Linux root file system. We don't use **boot.img** as well as **system.img** but download Linux kernel image and **rootsystem.img** instead.

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

Permanent link:
**http://wiki.odroid.com/odroid-c1/software/building_android?rev=1501574496**

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