**Build and Boot U-Boot and Linux on a Raspberry Pi 3 Model B+**

[Linux](https://www.thegoodpenguin.co.uk/blog/category/tech-blog/linux/) [Tech Blog](https://www.thegoodpenguin.co.uk/blog/category/tech-blog/)

With more than [30 million](https://twitter.com/EbenUpton/status/1205646606504275968) units in existence, the [Raspberry Pi](https://www.raspberrypi.org/) provides for an accessible and low-cost way to play with an [ARMv8](https://en.wikipedia.org/wiki/ARM_architecture#64/32-bit_architecture) embedded device using the latest and greatest upstream sources. In this post we’ll walk through the steps required to build U-Boot, the Linux kernel and a filesystem, as well as the steps required for preparing an SD card and booting it.

Let’s start by getting a toolchain that will build 64bit ARMv8 binaries, we’ll grab the latest [Linaro toolchain](https://releases.linaro.org/components/toolchain/binaries/latest-7/aarch64-linux-gnu/), unpack it and temporarily put it in our system path:

$ wget https://releases.linaro.org/components/toolchain/binaries/latest-7/aarch64-linux-gnu/gcc-linaro-7.5.0-2019.12-x86\_64\_aarch64-linux-gnu.tar.xz

$ tar -xf gcc-linaro-7.5.0-2019.12-x86\_64\_aarch64-linux-gnu.tar.xz -C ~/tools

$ export PATH=~/tools/gcc-linaro-7.5.0-2019.12-x86\_64\_aarch64-linux-gnu/bin:$PATH

Even though the Raspberry Pi can [directly boot a kernel](https://www.raspberrypi.org/documentation/configuration/config-txt/boot.md), we’d still like to boot via U-Boot. Thus let’s grab the latest version of mainline U-Boot, configure it for our Raspberry Pi and build a u-boot.bin binary:

$ git clone ~~git://git.denx.de/u-boot.git~~ https://source.denx.de/u-boot/u-boot.git # v2021.01-rc2-121-g5b8991c667f7

$ cd u-boot

$ make ARCH=arm CROSS\_COMPILE=aarch64-linux-gnu- rpi\_arm64\_defconfig

$ make ARCH=arm CROSS\_COMPILE=aarch64-linux-gnu- -j$(nproc)

$ cd ..

make ARCH=arm CROSS\_COMPILE=aarch64-linux-gnu- -j$(nproc) để tạo ra u-boot.bin

make menuconfig để chỉnh lại các config (support ethernet)