

Odroid C2 Real Time Kernel

The standard Linux kernel only meets soft real-time requirements: it provides basic POSIX operations for userspace time handling but has no guarantees for hard timing deadlines. With Ingo Molnar's Realtime Preemption patch (referenced to as RT-Preempt in this document) and Thomas Gleixner's generic clock event layer with high resolution support, the kernel gains hard realtime capabilities.

The RT-Preempt patch has raised quite some interest throughout the industry. Its clean design and consequent aim towards mainline integration makes it an interesting option for hard and firm realtime applications, reaching from professional audio to industrial control.

Goals of RT kernel

```
Making in-kernel locking-primitives (using spinlocks) preemptible though
reimplementation with rtmutexes:
  Critical sections protected by i.e. spinlock_t and rwlock_t are now
  preemptible.
  Implementing priority inheritance for in-kernel mutexes, spinlocks and
  rw_semaphores.
  Converting interrupt handlers into preemptible kernel threads
  Converting the old Linux timer API into separate infrastructures for high
  resolution kernel timers plus one for timeouts,
  leading to userspace POSIX timers with high resolution.
```

Build RT kernel

To build RT kernel we need to update the kernel with Real time kernel patches.

RT Patches file can be found on <https://www.kernel.org/pub/linux/kernel/projects/rt/>

Latest Patches for 3.14.x can be found on <https://www.kernel.org/pub/linux/kernel/projects/rt/3.14/>

Checkout

You can checkout Linux kernel source tree from [Hardkernel's Github](#).

RT Linux Kernel

Note: RT patches depend on the kernel version, so choose the relevant patch series.

```
$ git clone --depth 1 -b odroidc2-3.14.y
https://github.com/hardkernel/linux.git odroidc2-3.14.y-rt
$ cd odroidc2-3.14.y-rt
$ wget
```

```
https://www.kernel.org/pub/linux/kernel/projects/rt/3.14/older/patch-3.14.65-rt68.patch.gz
$ zcat patch-3.14.65-rt68.patch.gz > patch-3.14.65-rt68.patch
$ patch -p1 < patch-3.14.65-rt68.patch
```

Note: Arc arm64 rt kernel have some missing patches so we have to update the kernel for Following patches need to be applied to build the kernel.

Note: These patches can be found in 3.18.y-rt patch series.

```
arm64: Mark PMU interrupt IRQF_NO_THREAD.patch
arm64: Allow forced irq threading.patch
arch/arm64: Add lazy preempt support.patch
arm64: replace read_lock to rcu lock in call_step_hook
```

Note: Few file have conflict so you need to update manually these patches before build the kernel.

Note: Please find the links for all the updated patch in the following repository.

<https://github.com/moonlinux/linux/tree/odroidc2-3.14.y-rt>

BuildRT Linux Kernel

Kernel Configuration for RT Full Preemptible Kernel.

Select following options to build the kernel.

```
Kernel Features --->
  Preemption Model (Fully Preemptible Kernel (RT)) --->
    (X) Fully Preemptible Kernel (RT) --->
```

Note: update the odroidc2_defconfig with current modified.

```
$ cp .config arch/arm64/configs/odroidc2_defconfig
```

Then, Update your ODRROID's kernel.

You must do kernel configuration for ODRROID-C2, then start to build. Adding -j option on make command will help you to finish compiling faster.

```
$ make odroidc2_defconfig
$ make <-j4> Image dtbs modules
```

You have done to compile the Linux kernel (Image), the device tree file (.dtb) and kernel modules (.ko).

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 Image and DT(meson64_odroidc2.dtb) to the FAT partition(1st partition) in the Boot-Device.

```
$ mkdir -p mount
$ sudo mount /dev/sdc1 ./mount
$ sudo cp arch/arm64/boot/Image arch/arm64/boot/dts/meson64_odroidc2.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=arm64 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-c2/software/building_rt_kernel

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

