Support of the Nezha Allwinner D1 in meta-riscv

Prepared by Cezary Sobczak

Introduction

Some facts:

- Junior Embedded System Developer at 3mdeb - 2 years of experience.
- Student in the last year of master's at the Gdańsk University of Technology with specialties in Embedded Systems and Autonomous Vehicles.
- My main interests are automotive, IoT, embedded systems, and microcontrollers.

Agenda

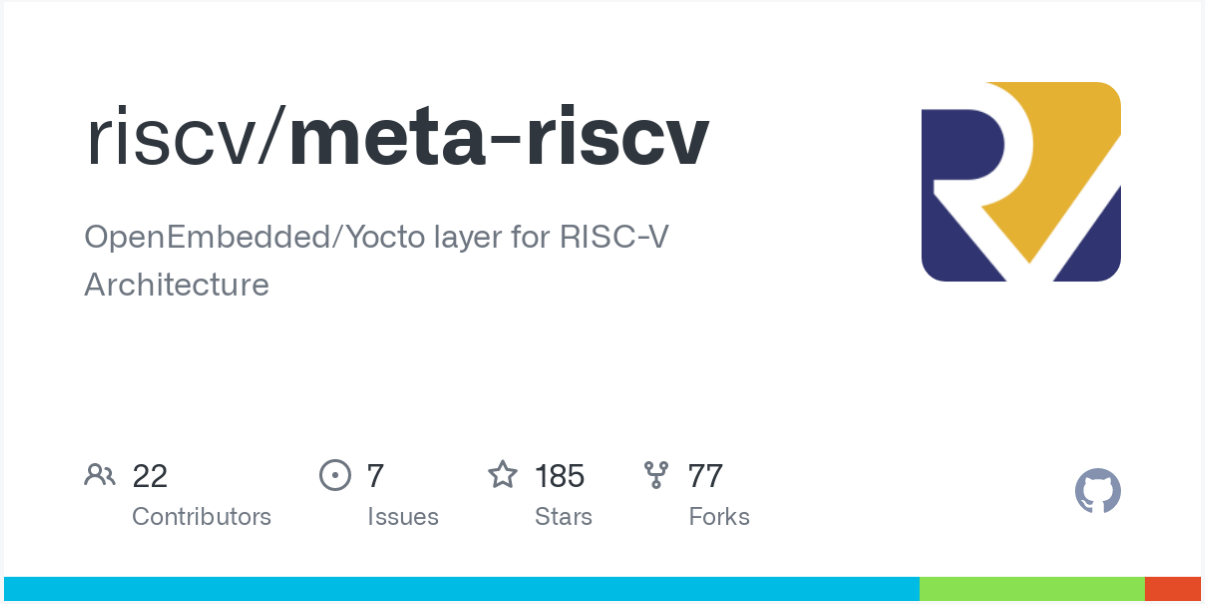
- Project backgroud
- What is a meta-riscv?
- Nezha board introduction
- Machine configuration
- Boot flow
- SD Card storage layout
- Preparing recipe for boot0
- Patches for OpenSBI
- U-Boot recipe adaptation
- Linux recipe adaptation
- Build minimal image
- Demo
- Known issues & needs

Project background

- Willingness to learn about RISC-V architecture as a part of master's thesis.
- At first, the plan was to create a simple OS for the BeagleV board.
- Market research for other SBCs or processors based at RISC-V ISA.
- SBC Nezha appears on the market with basic support of Linux base systems.
- Nezha board didn't have support for Yocto Project.

What is a meta-riscv?

meta-riscv is a OpenEmbedded / a Yocto layer for RISC-V-based boards and it contains a BSP for it. Here you can find machine configurations, recipes for specific firmware/software of the board, and examples of how to build the basic Yocto image for these machines eg. using kas.

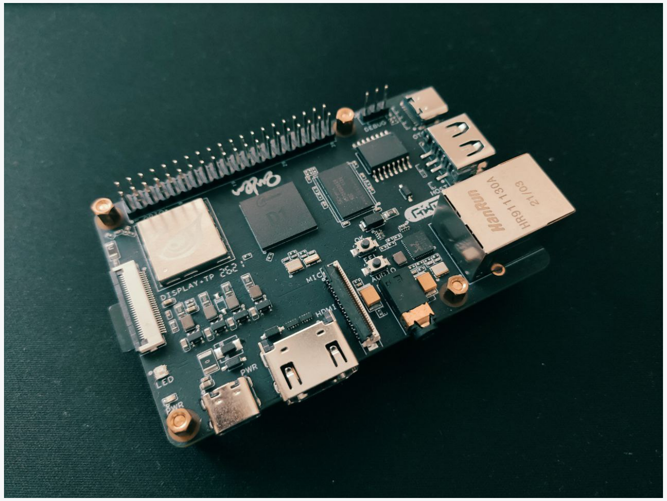


riscv/**meta-riscv**

OpenEmbedded/Yocto layer for RISC-V Architecture

22 Contributors · 7 Issues · 185 Stars · 77 Forks

Nezha board introduction



Nezha board is a development board that is designed by an AWOL. This project uses a D1 SoC from Allwinner which is used for the first time by the general public. Probably this board is the **first massive** produced and available SBC based at RISC-V architecture taking in mind a fact that BeagleV™ pilot program with version beta of the board was canceled in August 2021.

Specification of Allwinner D1 SoC

Name	Parameter description
XuanTie C906	Single core 1.0GHz 64-bit RISC-V processor
HiFi4 DSP	Cadence® Tensilica® HiFi 4
G2D 2D	graphics accelerators
DDR3 RAM	three variants - 512MB, 1GB or 2GB
SPI NAND	256MB of flash memory

Nezha board introduction

Periphelials

Name	Parameter description
Storage	Onboard 256MB spi-nand, support USB external U disk and SD card to expand storage
Network	Support Gigabit Ethernet, support 2.4GHz WiFi and Bluetooth, onboard antenna
Display	Support MIPI-DSI+TP screen interface, support HDMI output, support SPI screen
Audio	Microphone daughter board interface * 1, 3.5mm headphone jack * 1 (CTIA)
USB	power type-c, OTG type-c, HOST type-a
GPIO HEADER	Raspberry Pi like header
DEBUG	Dedicated header for serial communication (UART)

Machine configuration

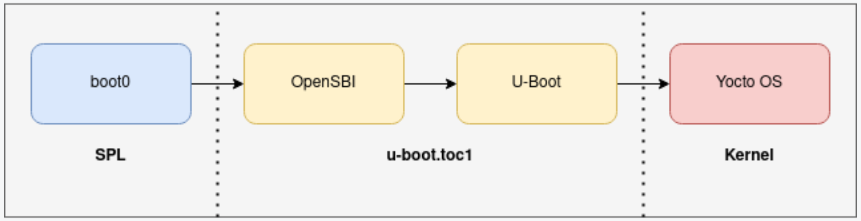
Machine configuration for boards can be found in **meta-riscv** at path conf/machine. Here you can check and change configuration for particular machine. Key features of **nezha-allwinner-d1.conf**:

- install kernel **fitImage** format
- install **boot0**, **U-Boot** and **OpenSBI**
- use **linux-nezha-dev** and **u-boot-nezha** as a virtual preferred provider
- U-Boot isn't set as the SBI payload, because D1 SoC using the U-Boot TOC1 image instead
- **KERNEL_DEVICETREE** and **RISCV_SBI_FDT** aren't set because the DTB is loaded from RAM at address **${fdtcontroladdr}**
- set proper U-Boot defconfig, entrypoint, device tree load address and binary file name:

```
UBOOT_MACHINE = "nezha_defconfig"
UBOOT_ENTRYPOINT = "0x40200000"
UBOOT_DTB_LOADADDRESS = "0x4FA00000"
UBOOT_DTB_BINARY ?= "sun20i-d1-nezha.dtb"
```

Boot flow

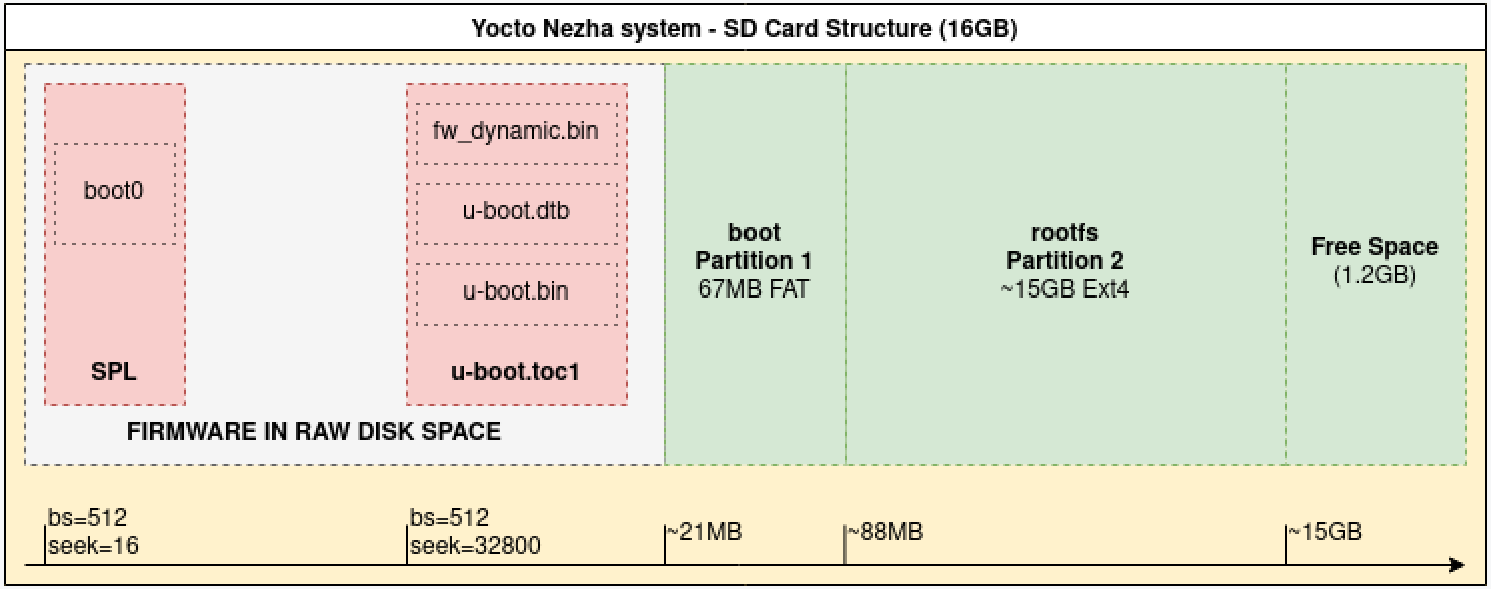
Boot firmware on D1 consists of three parts, which largely correspond to the components used by 64-bit ARM SoCs:



- boot0 - it is modified for this board and used as SPL due to features such as enabling the T-HEAD ISA and MMU extensions. Used instead of U-Boot SPL.
- OpenSBI - supervisor which is an interface between two less privileged modes boot0 and TPL bootloader.
- U-Boot - TPL bootloader which initializes additional hardware and loads kernel from storage or the network.

More information can be found at linux-sunxi wiki.

SD Card storage layout


Yocto Nezha system - SD Card Structure (16GB)
boot0
SPL
fw_dynamic.bin
u-boot.dtb
u-boot.bin
u-boot.toc1
FIRMWARE IN RAW DISK SPACE
boot Partition 1 67MB FAT
rootfs Partition 2 ~15GB Ext4
Free Space (1.2GB)
bs=512 seek=16
bs=512 seek=32800
~21MB
~88MB
~15GB

In meta-riscv you can find a formal description of the structure in nezha.wks file.

Preparing recipe for boot0

To fit in the Yocto Project build system some adjustments to boot0 Makefile had to be made:

- Allow overriding the variable which contains information about the used tool eg. C compiler and linker,
- Remove nostdinc from config.mk which helps build on different kinds of toolchains
- Fix build with **binutils v2.28** - it was necessary due to the new ISA specification version 20191213

Link to the recipe:

https://github.com/riscv/meta-riscv/blob/master/recipes-bsp/boot0/boot0.bb

Patches for OpenSBI

Mainline OpenSBI supports the C906 out of the box, but it needs a few tweaks and a new reset driver for the sunxi watchdog. Here we have two patches prepared by **Samuel Holland** which are applied during build process:

- Add a separate compatible timer for the D1 CLINT which does not support 64-bit MMIO access
- FDT requires match data to be constant. Match data stores hardware attributes that do not change at runtime, so it does not need to be mutable

Link to recipe:

https://github.com/riscv/meta-riscv/blob/master/recipes-bsp/opensbi/opensbi_%25.bbappend

U-Boot recipe adaptation

Stand-alone recipe only for Nezha board was prepared. Main changes compared to mainline U-Boot recipe:

- Use a patched version of 2022.01 version of U-Boot from **Fu Wei** (Fedora) fork:

  ```
  SRC_URI = "git://github.com/tekkamanninja/u-boot.git;branch=allwinner_d1"
  SRCREV = "6db9960b2443ef84b88a573cb5817f8e0ef3712e"
  ```

- Apply a patch that fixes a problem during uncompressing the Kernel Image:

  ```
  Error: inflate() returned -5
  Image too large: increase CONFIG_SYS_BOOTM_LEN
  ```

- Fix build with **binutils v2.28** - it was necessary due to the new ISA specification version 20191213

U-Boot recipe adaptation

- Provide toc.cfg used by mkimage tool to create TOC1 image

```
[opensbi]
file = fw_dynamic.bin
addr = 0x40000000
[dtb]
file = u-boot.dtb
addr = 0x44000000
[u-boot]
file = u-boot.bin
addr = 0x4a000000
```

- Provide custom U-Boot Environment file uEnv-nezha.txt

```
bootargs=earlycon=sbi clk_ignore_unused initcall_debug=0 console=ttyS0,115200
        loglevel=8 root=/dev/mmcblk0p2 rootwait  init=/sbin/init
bootcmd_load_f=load ${devtype} ${devnum}:${distro_bootpart} ${ramdisk_addr_r} fitImage
bootcmd_run=bootm ${ramdisk_addr_r} - ${fdtcontroladdr}
bootcmd=run bootcmd_load_f; run bootcmd_run
```

U-Boot recipe adaptation

- Add new task do_make_toc1_image which is executed after do_compile and before do_deploy. It has dependency at OpenSBI do_deploy task too

```
do_make_toc1_image() {
  cd ${B}
  cp ${DEPLOY_DIR_IMAGE}/fw_dynamic.bin ${B}
  ${B}/tools/mkimage -T sunxi_toc1 -d ${WORKDIR}/toc.cfg ${B}/u-boot.toc1
}
```

Link to recipe:

https://github.com/riscv/meta-riscv/blob/master/recipes-bsp/u-boot/u-boot-nezha.bb

Linux recipe adaptation

Same as for U-Boot, a stand-alone recipe was prepared. The new recipe provides the following things:

- Current version of kernel: `v5.16`
- Use a Fu Wei (Fedora) fork with support of the Allwinner D1 chip

```
SRCREV_meta ?= "ea948a0983d7b7820814e5bce4eda3079201bd95"
SRCREV_machine ?= "af3f4a1caec12845b809fba959e6334ab3b52a40"
FORK ?= "tekkamanninja"
BRANCH ?= "allwinner_nezha_d1_devel"
KMETA = "kernel-meta"
```

Linux recipe adaptation

Same as for U-Boot, a stand-alone recipe was prepared. The new recipe provides the following things:

- Add **cgroups** and **autofs4** kernel features with the following:

```
KERNEL_FEATURES += "features/cgroups/cgroups.cfg"
KERNEL_FEATURES += "ktypes/standard/standard.cfg"
```

- Fix build with **binutils v2.28** - it was necessary due to the new ISA specification version 20191213

Link to recipe:

https://github.com/riscv/meta-riscv/blob/master/recipes-kernel/linux/linux-nezha-dev.bb

Build minimal image

Before building you should have the following things on your host PC:

- Install **kas-container**

```
$ sudo wget https://raw.githubusercontent.com/siemens/kas/master/kas-container \
-O /usr/bin/kas-container
$ sudo chmod 755 /usr/bin/kas-container
```

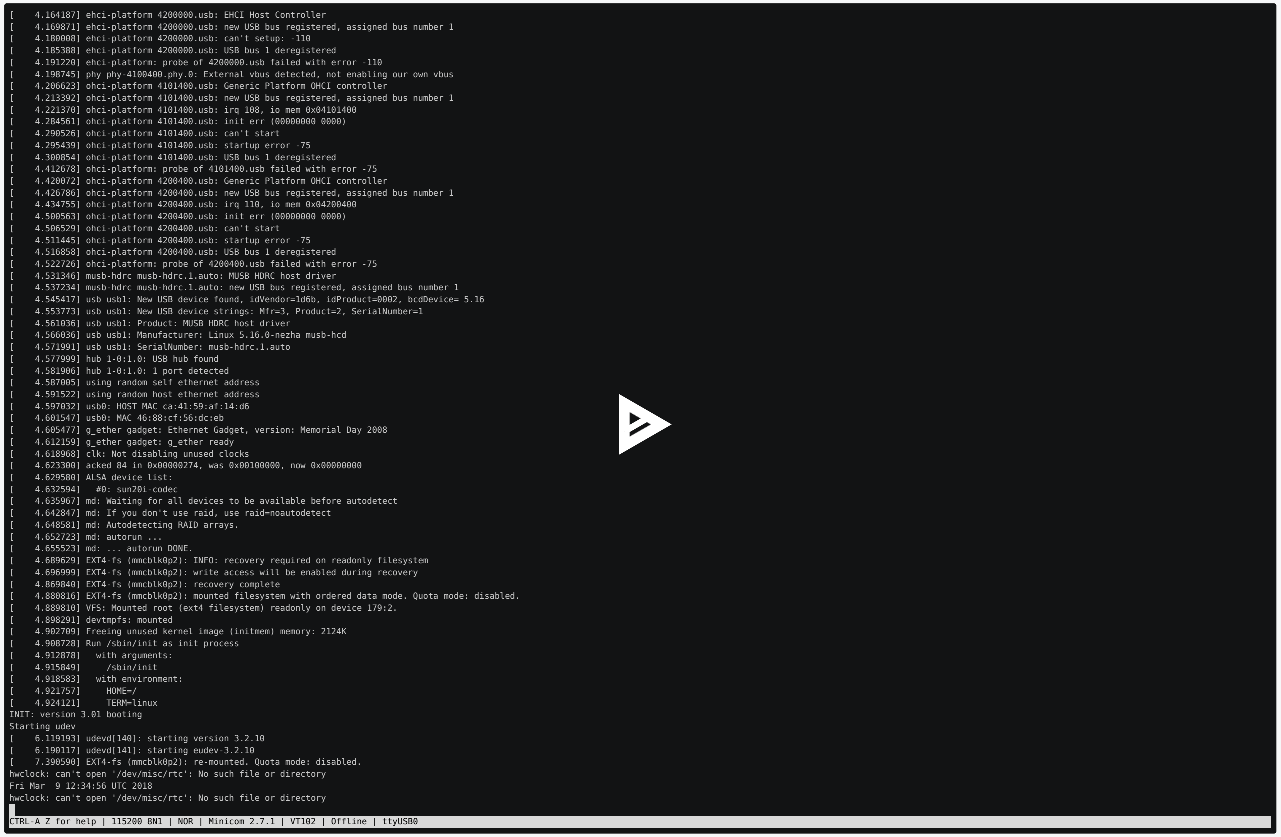
- Clone **meta-riscv** repository

```
$ git clone https://github.com/riscv/meta-riscv.git
```

To build core-image-minimal Poky distro run the following command:

```
$ SHELL=/bin/bash kas-container build meta-riscv/nezha.yml
```

Demo



```
[    4.164187] ehci-platform 4200000.usb: EHCI Host Controller
[    4.169871] ehci-platform 4200000.usb: new USB bus registered, assigned bus number 1
[    4.180008] ehci-platform 4200000.usb: can't setup: -110
[    4.185388] ehci-platform 4200000.usb: USB bus 1 deregistered
[    4.191220] ehci-platform: probe of 4200000.usb failed with error -110
[    4.198745] phy phy-4100400.phy.0: External vbus detected, not enabling our own vbus
[    4.206623] ohci-platform 4101400.usb: Generic Platform OHCI controller
[    4.213392] ohci-platform 4101400.usb: new USB bus registered, assigned bus number 1
[    4.221370] ohci-platform 4101400.usb: irq 108, io mem 0x04101400
[    4.284561] ohci-platform 4101400.usb: init err (00000000 0000)
[    4.290526] ohci-platform 4101400.usb: can't start
[    4.295439] ohci-platform 4101400.usb: startup error -75
[    4.300854] ohci-platform 4101400.usb: USB bus 1 deregistered
[    4.412678] ohci-platform: probe of 4101400.usb failed with error -75
[    4.420072] ohci-platform 4200400.usb: Generic Platform OHCI controller
[    4.426786] ohci-platform 4200400.usb: new USB bus registered, assigned bus number 1
[    4.434755] ohci-platform 4200400.usb: irq 110, io mem 0x04200400
[    4.500563] ohci-platform 4200400.usb: init err (00000000 0000)
[    4.506529] ohci-platform 4200400.usb: can't start
[    4.511445] ohci-platform 4200400.usb: startup error -75
[    4.516858] ohci-platform 4200400.usb: USB bus 1 deregistered
[    4.522726] ohci-platform: probe of 4200400.usb failed with error -75
[    4.531346] musb-hdrc musb-hdrc.1.auto: MUSB HDRC host driver
[    4.537234] musb-hdrc musb-hdrc.1.auto: new USB bus registered, assigned bus number 1
[    4.545417] usb usb1: New USB device found, idVendor=1d6b, idProduct=0002, bcdDevice= 5.16
[    4.553773] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[    4.561036] usb usb1: Product: MUSB HDRC host driver
[    4.566036] usb usb1: Manufacturer: Linux 5.16.0-nezha musb-hcd
[    4.571991] usb usb1: SerialNumber: musb-hdrc.1.auto
[    4.577999] hub 1-0:1.0: USB hub found
[    4.581906] hub 1-0:1.0: 1 port detected
[    4.587005] using random self ethernet address
[    4.591522] using random host ethernet address
[    4.597032] usb0: HOST MAC ca:41:59:af:14:d6
[    4.601547] usb0: MAC 46:88:cf:56:dc:eb
[    4.605477] g_ether gadget: Ethernet Gadget, version: Memorial Day 2008
[    4.612159] g_ether gadget: g_ether ready
[    4.618968] clk: Not disabling unused clocks
[    4.623300] acked 84 in 0x00000274, was 0x00100000, now 0x00000000
[    4.629580] ALSA device list:
[    4.632594]   #0: sun20i-codec
[    4.635967] md: Waiting for all devices to be available before autodetect
[    4.642847] md: If you don't use raid, use raid=noautodetect
[    4.648581] md: Autodetecting RAID arrays.
[    4.652723] md: autorun ...
[    4.655523] md: ... autorun DONE.
[    4.689629] EXT4-fs (mmcblk0p2): INFO: recovery required on readonly filesystem
[    4.696999] EXT4-fs (mmcblk0p2): write access will be enabled during recovery
[    4.869840] EXT4-fs (mmcblk0p2): recovery complete
[    4.880816] EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Quota mode: disabled.
[    4.889810] VFS: Mounted root (ext4 filesystem) readonly on device 179:2.
[    4.898291] devtmpfs: mounted
[    4.902709] Freeing unused kernel image (initmem) memory: 2124K
[    4.908728] Run /sbin/init as init process
[    4.912878]   with arguments:
[    4.915849]     /sbin/init
[    4.918583]   with environment:
[    4.921757]     HOME=/
[    4.924121]     TERM=linux
INIT: version 3.01 booting
Starting udev
[    6.119193] udevd[140]: starting version 3.2.10
[    6.190117] udevd[141]: starting eudev-3.2.10
[    7.390590] EXT4-fs (mmcblk0p2): re-mounted. Quota mode: disabled.
hwclock: can't open '/dev/misc/rtc': No such file or directory
Fri Mar  9 12:34:56 UTC 2018
hwclock: can't open '/dev/misc/rtc': No such file or directory

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB0
```

Known issues & needs

- **U-Boot SPL** - is currently available in some form:

  https://github.com/smaeul/u-boot/commit/7f9f2708f1b49f1936731aab4019cdff47b8dc29

- **rng-tools** - for some reason it crashes during the start with SIGSEGV in libc:

```
[   10.792295] rngd[139]: unhandled signal 11 code 0x2 at 0x0000003fc72e1378 in libc-2.35.so[3fc727e000+fd0
[   10.948096] CPU: 0 PID: 139 Comm: rngd Not tainted 5.16.0-nezha #1
```

  This problem doesn't exist when the haveged random number generator is used in the build.

- **WiFi & Bluetooth module** - for now it isn't possible to use a wireless interface. There is a need to port XR829 kernel module for version v5.16 and higher from Tina-Linux (kernel version: v5.4)

Q&A