



Hardware random overview



A quality version of this page, approved on 17 February 2021, was based off this revision.

This article gives information about the Linux[®] hardware random framework.

Contents

1 Article Purpose	3
2 Framework purpose	4
3 System overview	5
3.1 Component description	5
3.2 API description	6
4 Configuration	7
4.1 Kernel configuration	7
4.2 Device tree configuration	7
5 How to use the framework	8
5.1 How to use from char device	8
5.2 How to use from sysfs	8
6 How to trace and debug the framework	9
7 Source code location	10
8 To go further	11
9 References	12



1 Article Purpose

This article gives information about the hardware random (HWRNG) framework.

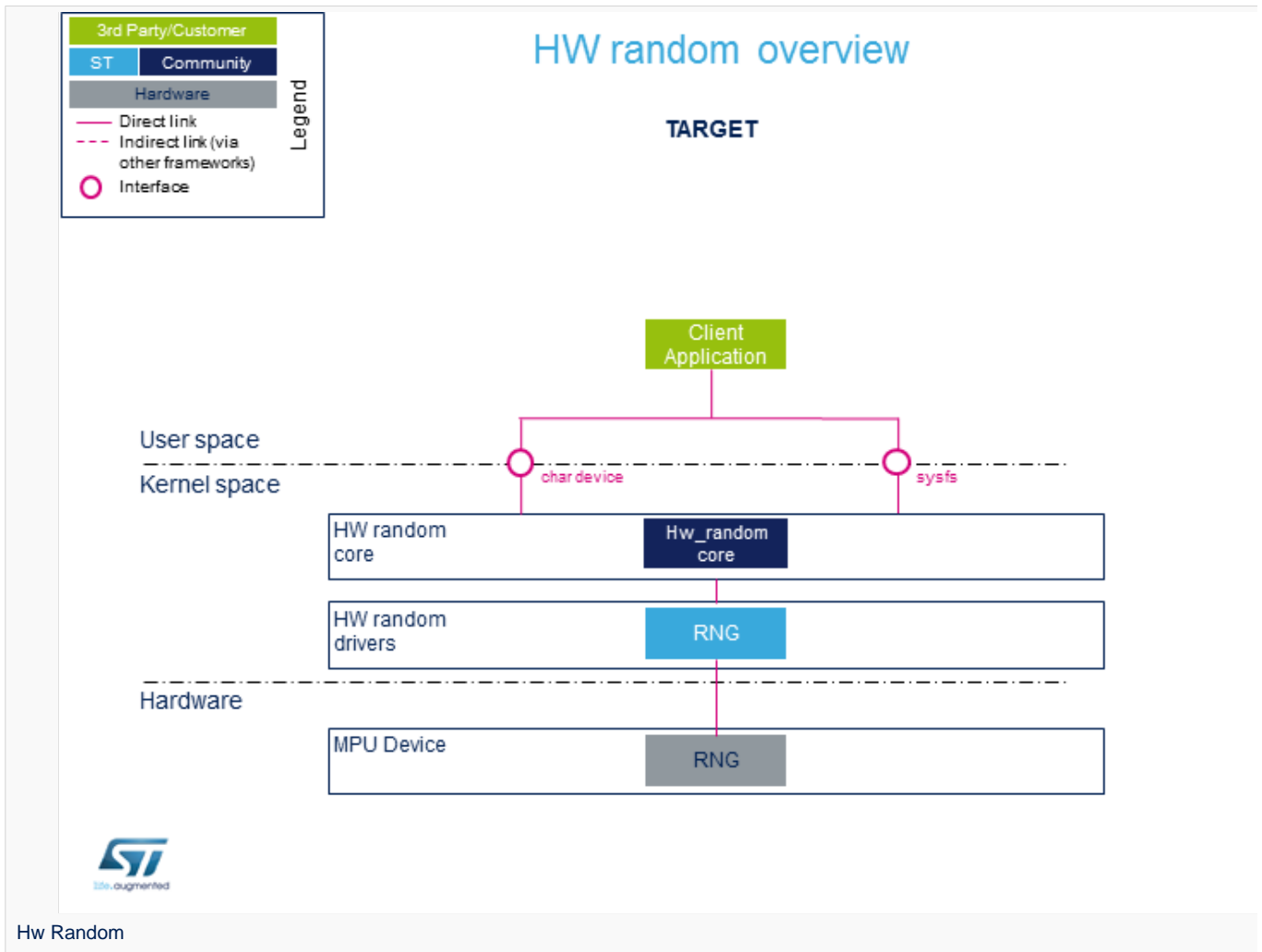


2 Framework purpose

The Hardware random framework is integrated in the kernel. It provides access to RNG peripherals and focuses on supporting the hardware number generator.

3 System overview

The HW random framework allows retrieving random numbers in userland.



3.1 Component description

- **HW random core** (Kernel space)

Generic interface in kernel space. This layer is in charge of creating the character device (char device) and sysfs to access hw_random.

- **RNG** (Kernel space)

Hardware random Linux[®] drivers handling the HW blocks.

- **RNG** (Hardware)

HW blocks handling the RNG peripheral.



3.2 API description

The Hardware random framework uses char device API^[1] ioctl operations. For additional information, refer to:

- sysfs interface.
- Kernel Documentation directory^[2]



4 Configuration

4.1 Kernel configuration

The Hardware random support is activated by default in ST deliveries. No specific configuration is required apart from enabling or disabling peripheral support using Linux® Menuconfig tool. Refer to [Menuconfig](#) or [how to configure kernel](#) and select:

```
[*] Device Drivers --->
  [*] Character devices --->
    [*] Hardware Random Number Generator Core support --->
      [*] STMicroelectronics STM32 random number generator
```

4.2 Device tree configuration

DT configuration can be done thanks to the [STM32CubeMX](#).

A detailed device tree configuration is described in [RNG device tree configuration](#).



5 How to use the framework

The framework provides external interfaces from userland : [How to control RNG](#).

5.1 How to use from char device

The community tool for using Hardware random framework is `rng_tools`^[3] which provides a complete set of utilities related to random number generators:

- **rngd**: runs a background daemon that opens `/dev/hwrng` file (default) to connect and retrieve random numbers.
- **rngtest**: runs different tests that check the entropy and verify the compliance regarding FIPS 140-2 standard.

5.2 How to use from sysfs

Available devices compatible with Hardware framework can be listed using `sysfs` commands:

```
Board $> cat /sys/class/misc/hw_random/rng_available  
stm32-rng
```

The selected device is shown here:

```
Board $> cat /sys/class/misc/hw_random/rng_current  
stm32-rng
```

To select a different device:

```
Board $> echo "stm32-rng"> /sys/class/misc/hw_random/rng_current
```




6 How to trace and debug the framework

Light information on the framework can be accessed by using `sysfs`.

By default, the framework does not provide any specific debug output or dynamic debugging tool.



7 Source code location

Hardware random drivers and framework are available here^[4].



8 To go further

Code examples are directly available from [rng-tools^{\[3\]}](#) github.



9 References

- <https://bootlin.com/doc/legacy/accessing-hardware/accessing-hardware.pdf>
- Documentation/admin-guide/hw_random.rst
- 3.03.1 Rng_tools source code
- drivers/char/hw_random , Hw_random sources

Linux[®] is a registered trademark of Linus Torvalds.

Random Number Generator

System File System (See <https://en.wikipedia.org/wiki/Sysfs> for more details)

Application programming interface

Device Tree