



Hardware random overview



Contents

1. Hardware random overview	3
2. How to control a RNG in userspace	12
3. Menuconfig or how to configure kernel	21
4. RNG device tree configuration	30
5. STM32CubeMX	39



A quality version of this page, approved on 11 February 2019, was based off this revision.

Template:ArticleMainWriter Template:ArticleApprovedVersion

SUMMARY

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

Contents

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

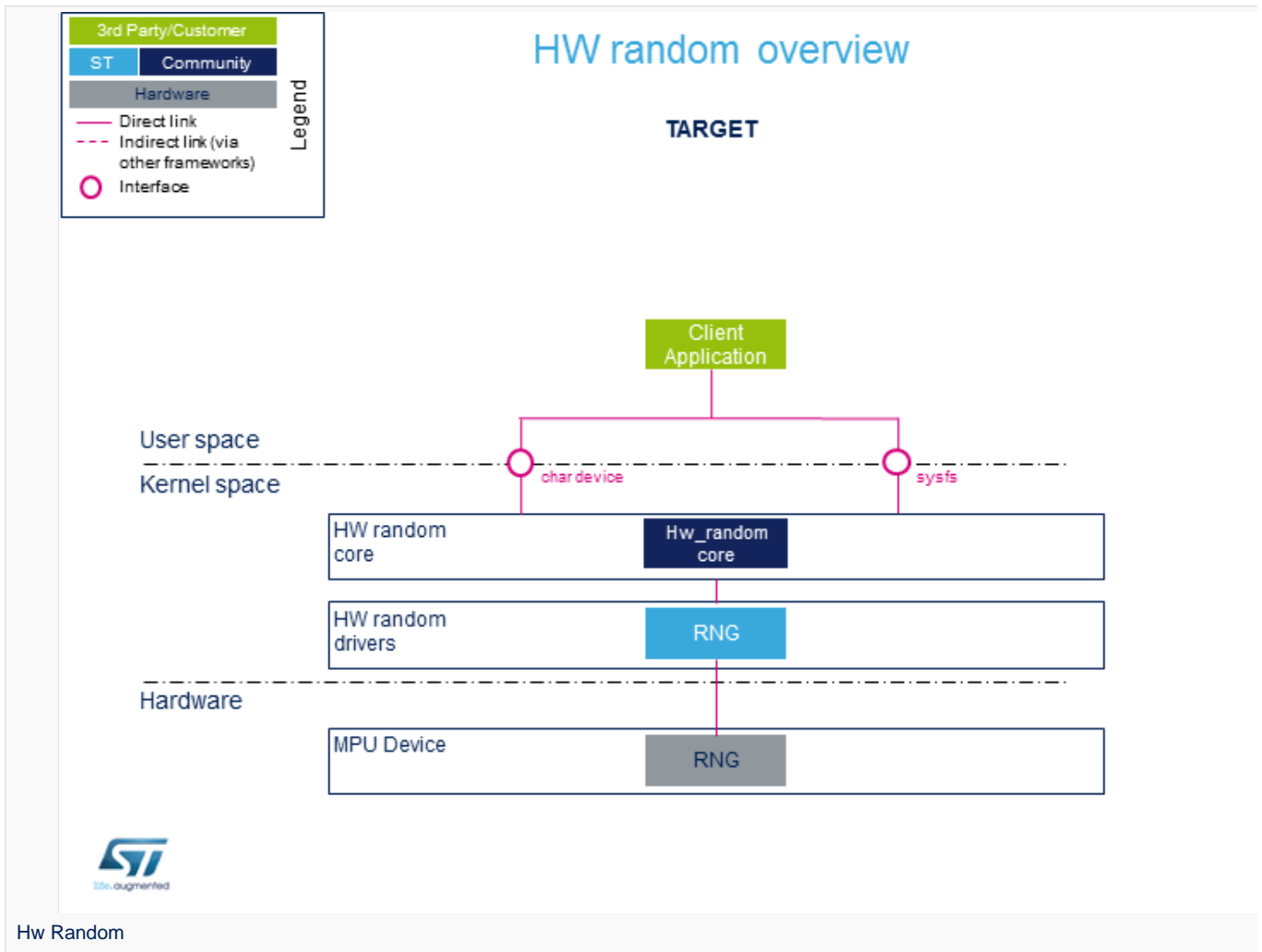


1 Framework purpose

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

2 System overview

The HW random framework allows retrieving random numbers in userland.



2.1 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.



2.2 API description

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

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



3 Configuration

3.1 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
```

3.2 Device tree configuration

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

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



4 How to use the framework

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

4.1 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.

4.2 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
```




5 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.



6 Source code location

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



7 To go further

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



8 References

- <https://bootlin.com/doc/legacy/accessing-hardware/accessing-hardware.pdf>
- Documentation/hw_random.txt
- 3.03.1 Rng_tools source code
- [drivers/char/hw_random . Hw_random sources](#)

Stable: 03.02.2020 - 08:42 / Revision: 03.02.2020 - 08:27

Template:ArticleMainWriter Template:ArticleApprovedVersion

SUMMARY

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

Contents

1 Framework purpose	13
2 System overview	14
2.1 Component description	14
2.2 API description	15
3 Configuration	16
3.1 Kernel configuration	16
3.2 Device tree configuration	16
4 How to use the framework	17
4.1 How to use from char device	17
4.2 How to use from sysfs	17
5 How to trace and debug the framework	18
6 Source code location	19
7 To go further	20
8 References	21

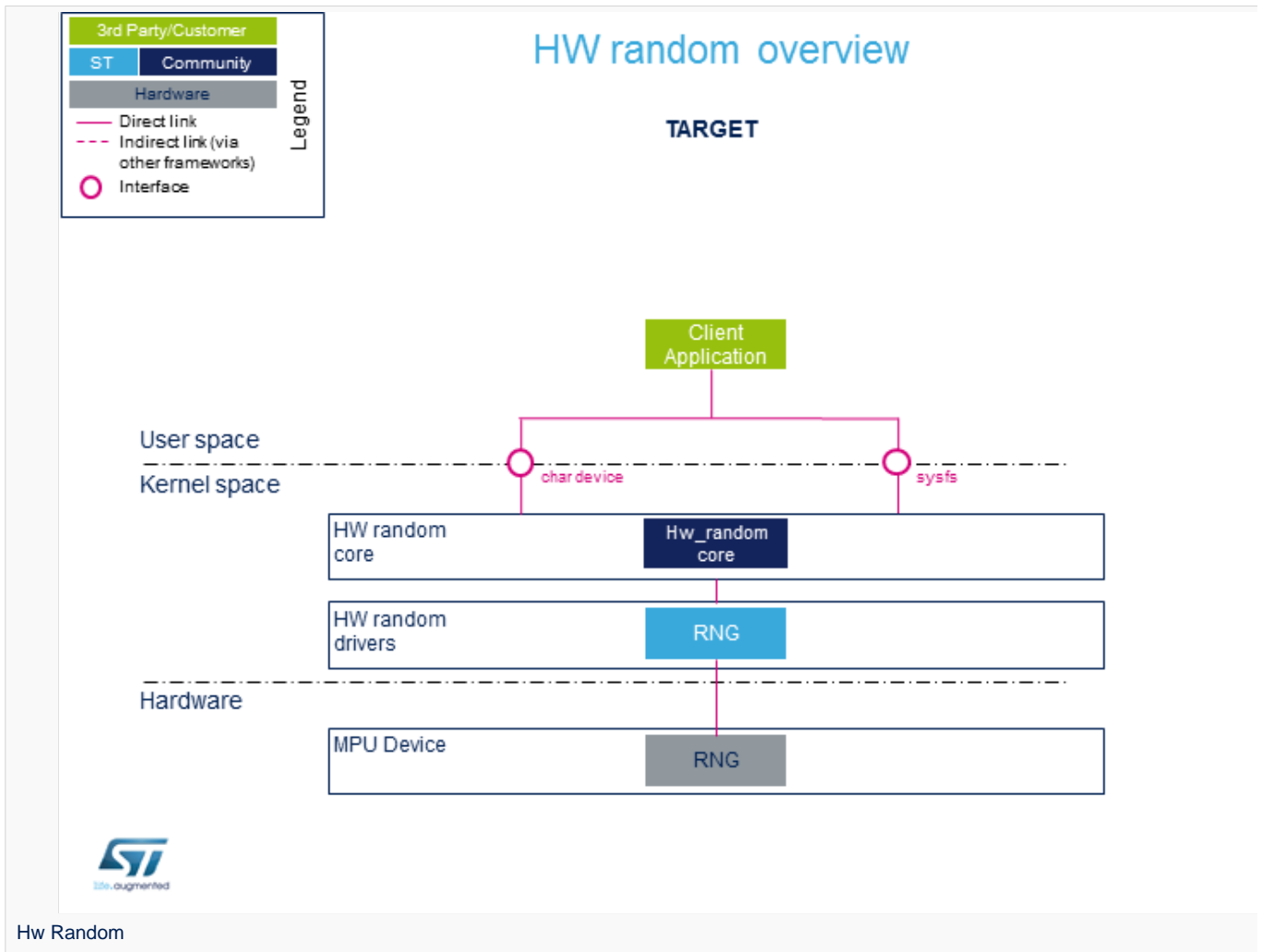


1 Framework purpose

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

2 System overview

The HW random framework allows retrieving random numbers in userland.



2.1 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.



2.2 API description

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

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



3 Configuration

3.1 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
```

3.2 Device tree configuration

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

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



4 How to use the framework

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

4.1 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.

4.2 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
```



5 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.



6 Source code location

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



7 To go further

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



8 References

- <https://bootlin.com/doc/legacy/accessing-hardware/accessing-hardware.pdf>
- Documentation/hw_random.txt
- 3.03.1 Rng_tools source code
- [drivers/char/hw_random . Hw_random sources](#)

Stable: 31.03.2021 - 08:47 / Revision: 26.03.2021 - 08:44

Template:ArticleMainWriter Template:ArticleApprovedVersion

SUMMARY

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

Contents

1 Framework purpose	22
2 System overview	23
2.1 Component description	23
2.2 API description	24
3 Configuration	25
3.1 Kernel configuration	25
3.2 Device tree configuration	25
4 How to use the framework	26
4.1 How to use from char device	26
4.2 How to use from sysfs	26
5 How to trace and debug the framework	27
6 Source code location	28
7 To go further	29
8 References	30

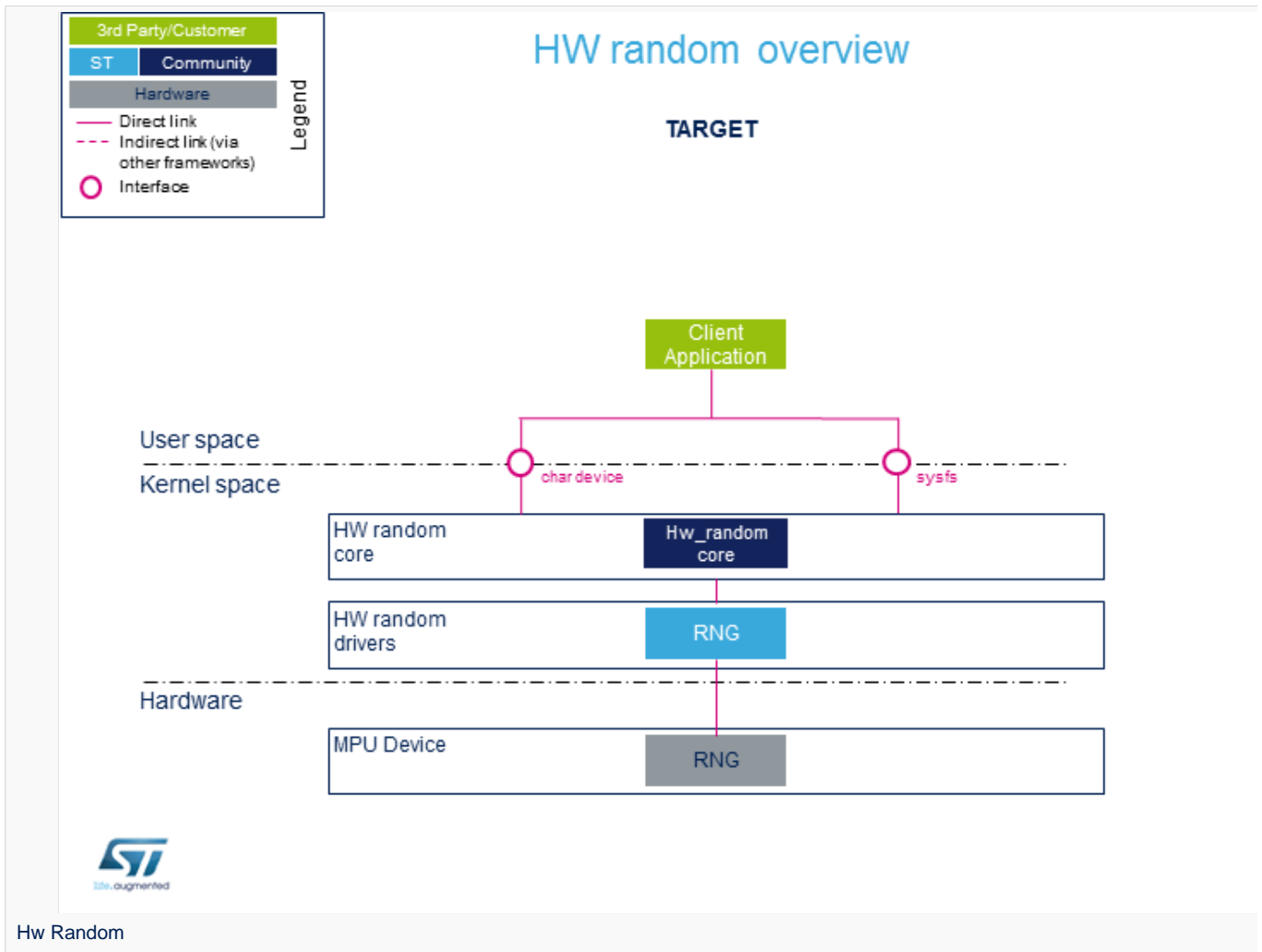


1 Framework purpose

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

2 System overview

The HW random framework allows retrieving random numbers in userland.



2.1 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.



2.2 API description

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

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



3 Configuration

3.1 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
```

3.2 Device tree configuration

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

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



4 How to use the framework

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

4.1 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.

4.2 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
```



5 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.



6 Source code location

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



7 To go further

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



8 References

- <https://bootlin.com/doc/legacy/accessing-hardware/accessing-hardware.pdf>
- Documentation/hw_random.txt
- 3.03.1 Rng_tools source code
- [drivers/char/hw_random . Hw_random sources](#)

Stable: 13.05.2020 - 08:40 / Revision: 13.05.2020 - 08:39

Template:ArticleMainWriter Template:ArticleApprovedVersion

SUMMARY

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

Contents

1 Framework purpose	31
2 System overview	32
2.1 Component description	32
2.2 API description	33
3 Configuration	34
3.1 Kernel configuration	34
3.2 Device tree configuration	34
4 How to use the framework	35
4.1 How to use from char device	35
4.2 How to use from sysfs	35
5 How to trace and debug the framework	36
6 Source code location	37
7 To go further	38
8 References	39

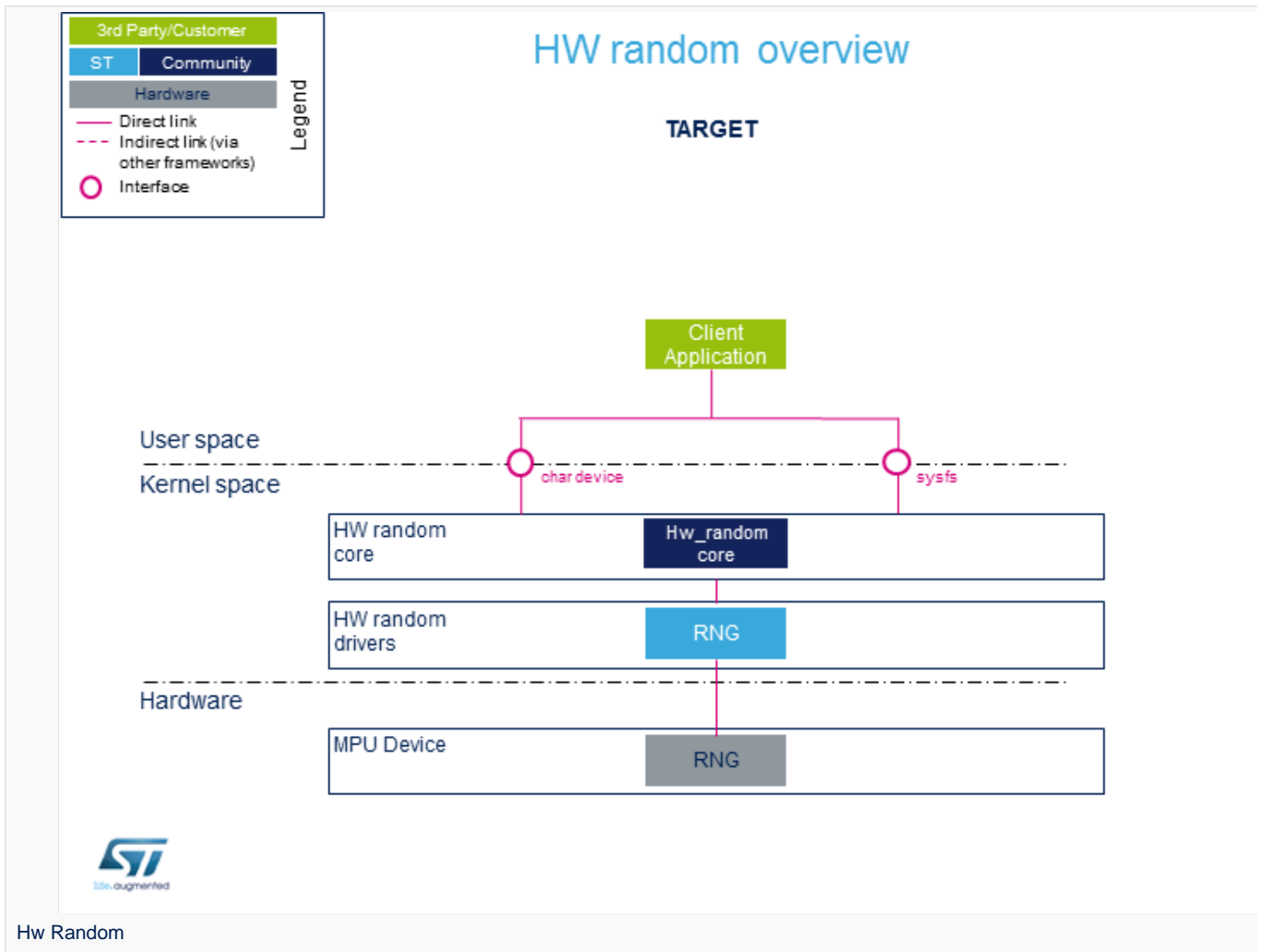


1 Framework purpose

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

2 System overview

The HW random framework allows retrieving random numbers in userland.



2.1 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.



2.2 API description

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

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



3 Configuration

3.1 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
```

3.2 Device tree configuration

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

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



4 How to use the framework

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

4.1 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.

4.2 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
```



5 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.



6 Source code location

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



7 To go further

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



8 References

- <https://bootlin.com/doc/legacy/accessing-hardware/accessing-hardware.pdf>
- Documentation/hw_random.txt
- 3.03.1 Rng_tools source code
- [drivers/char/hw_random . Hw_random sources](#)

Stable: 23.09.2020 - 13:22 / Revision: 12.06.2020 - 13:25

Template:ArticleMainWriter Template:ArticleApprovedVersion

SUMMARY

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

Contents

1 Framework purpose	40
2 System overview	41
2.1 Component description	41
2.2 API description	42
3 Configuration	43
3.1 Kernel configuration	43
3.2 Device tree configuration	43
4 How to use the framework	44
4.1 How to use from char device	44
4.2 How to use from sysfs	44
5 How to trace and debug the framework	45
6 Source code location	46
7 To go further	47
8 References	48

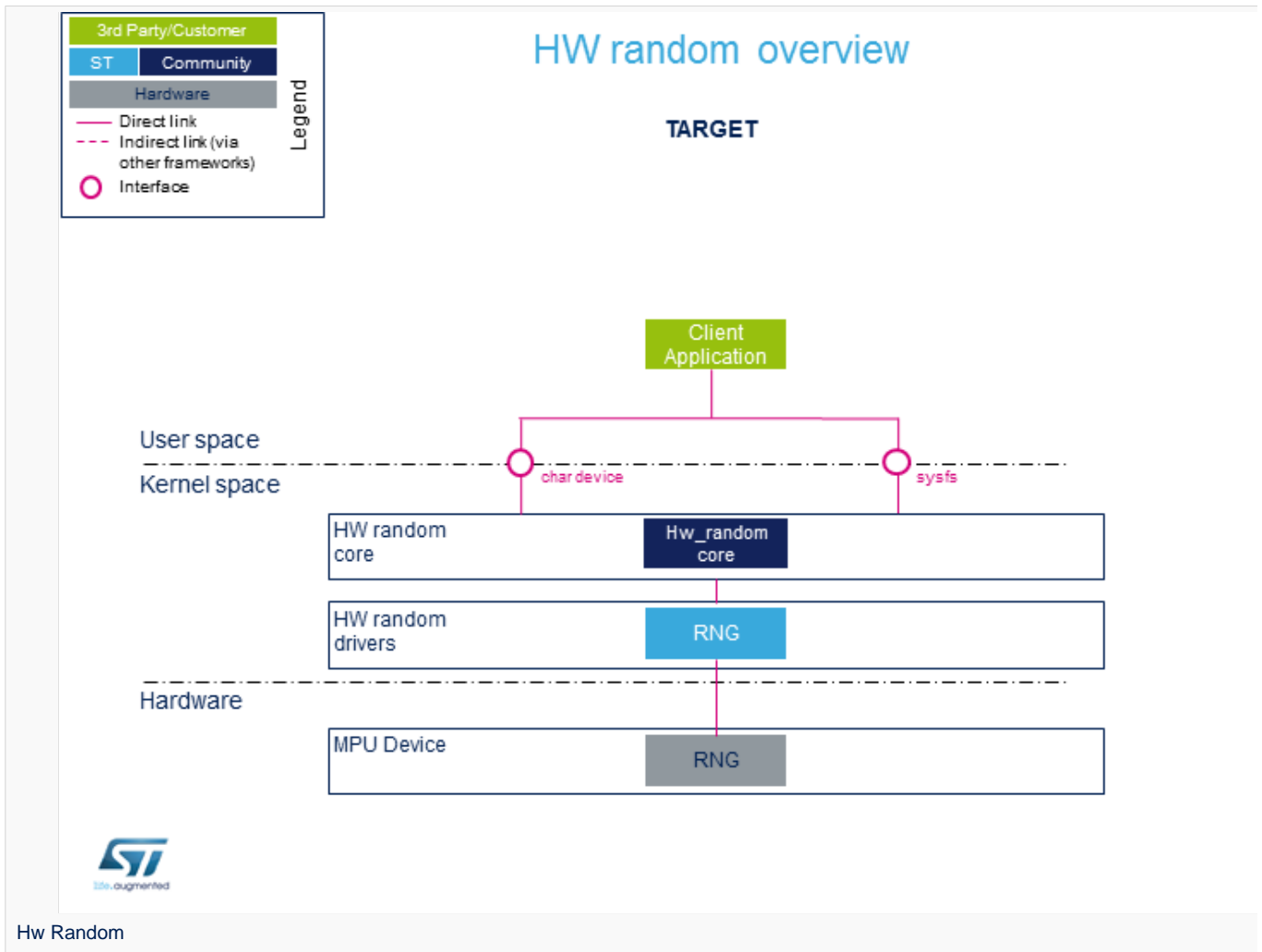


1 Framework purpose

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

2 System overview

The HW random framework allows retrieving random numbers in userland.



2.1 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.



2.2 API description

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

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



3 Configuration

3.1 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
```

3.2 Device tree configuration

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

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



4 How to use the framework

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

4.1 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.

4.2 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
```



5 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.



6 Source code location

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



7 To go further

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



8 References

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