Hardware random overview

Contents

Stable: 17.02.2021 - 19:55 / Revision: 17.02.2021 - 19:55

Stable: 17.02.2021 - 10:00 | Revision: 17.02.2021 - 10:00

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.

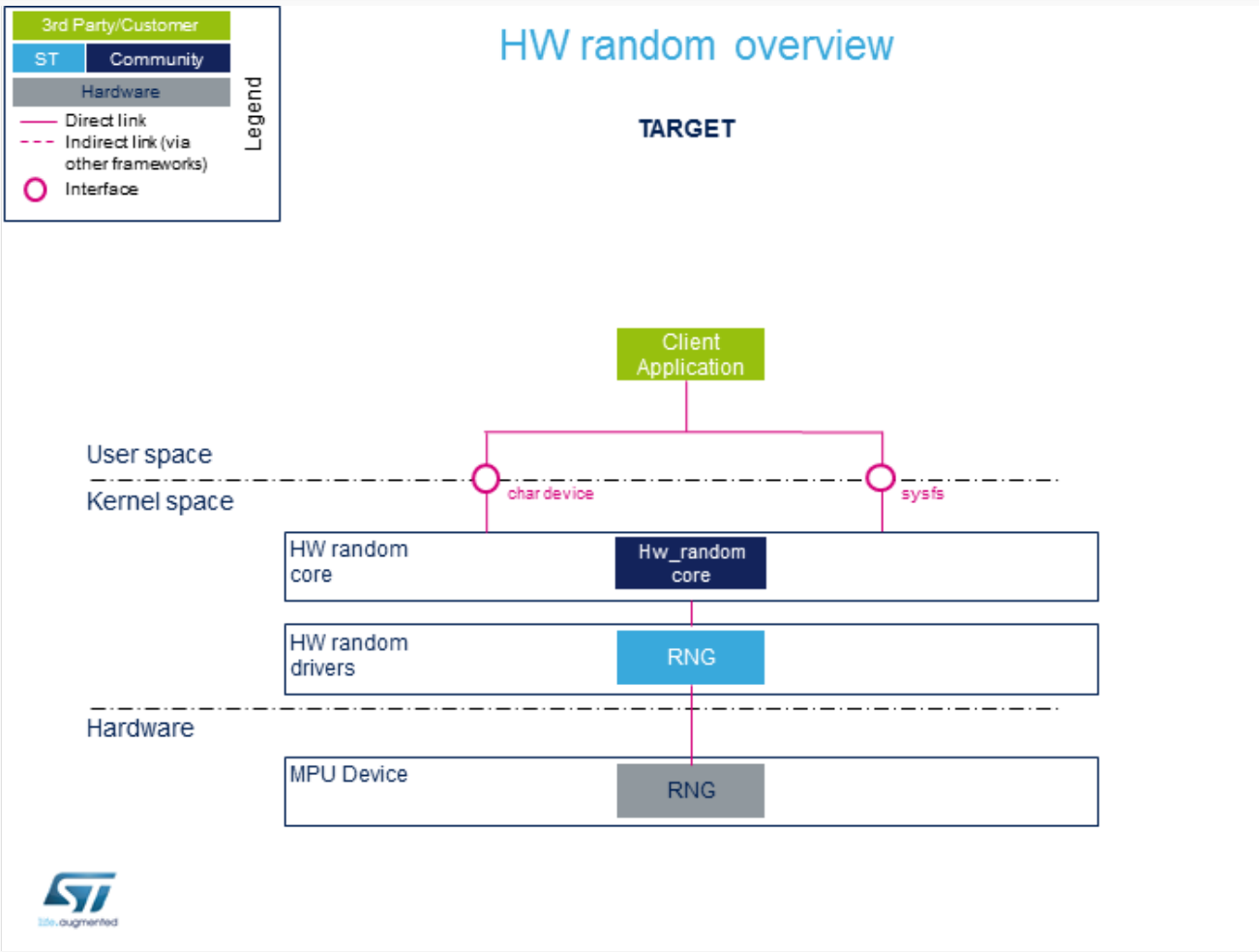
Contents

1 Framework purpose

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.



Hw Random

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.

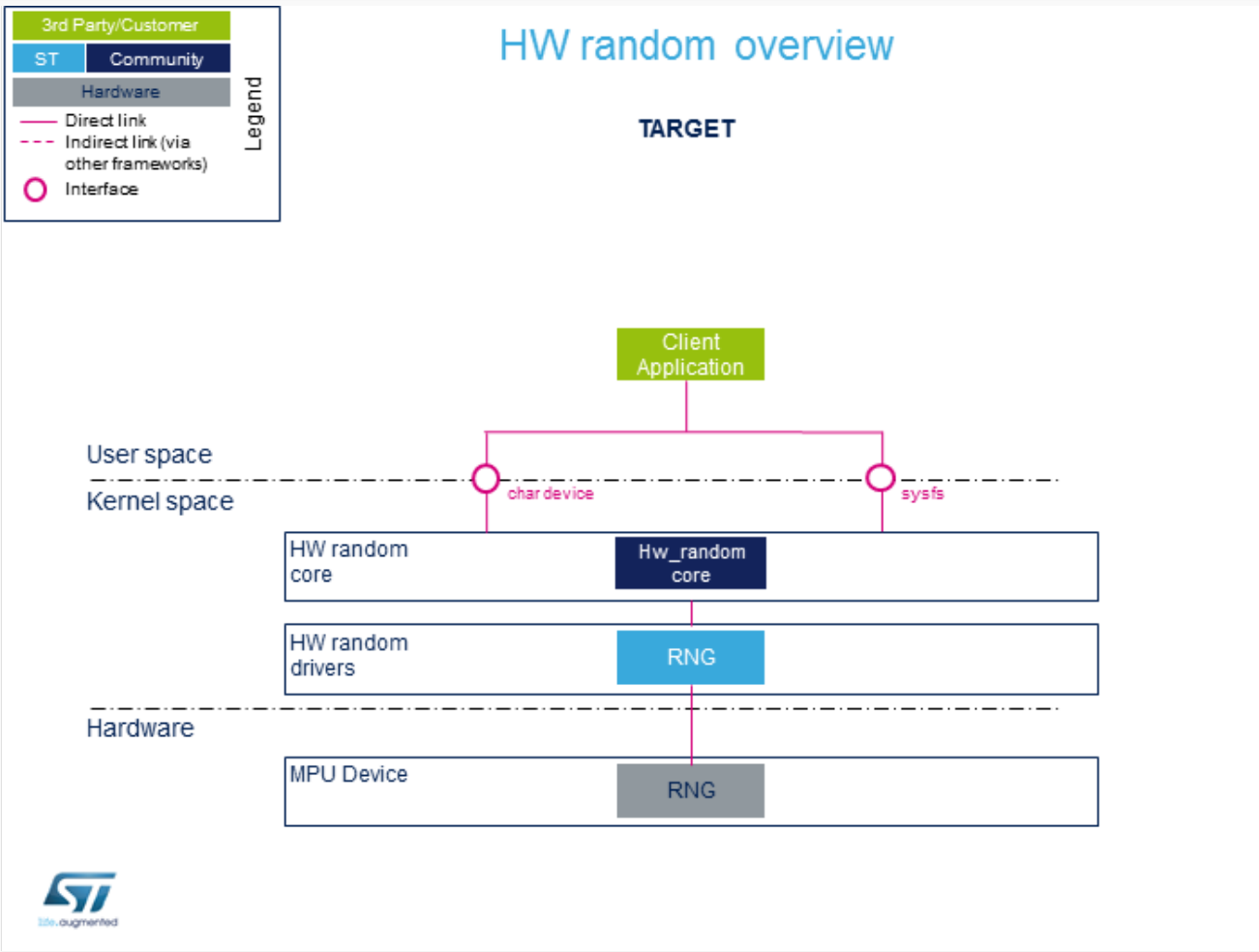
Contents

1 Framework purpose

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.



Hw Random

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.

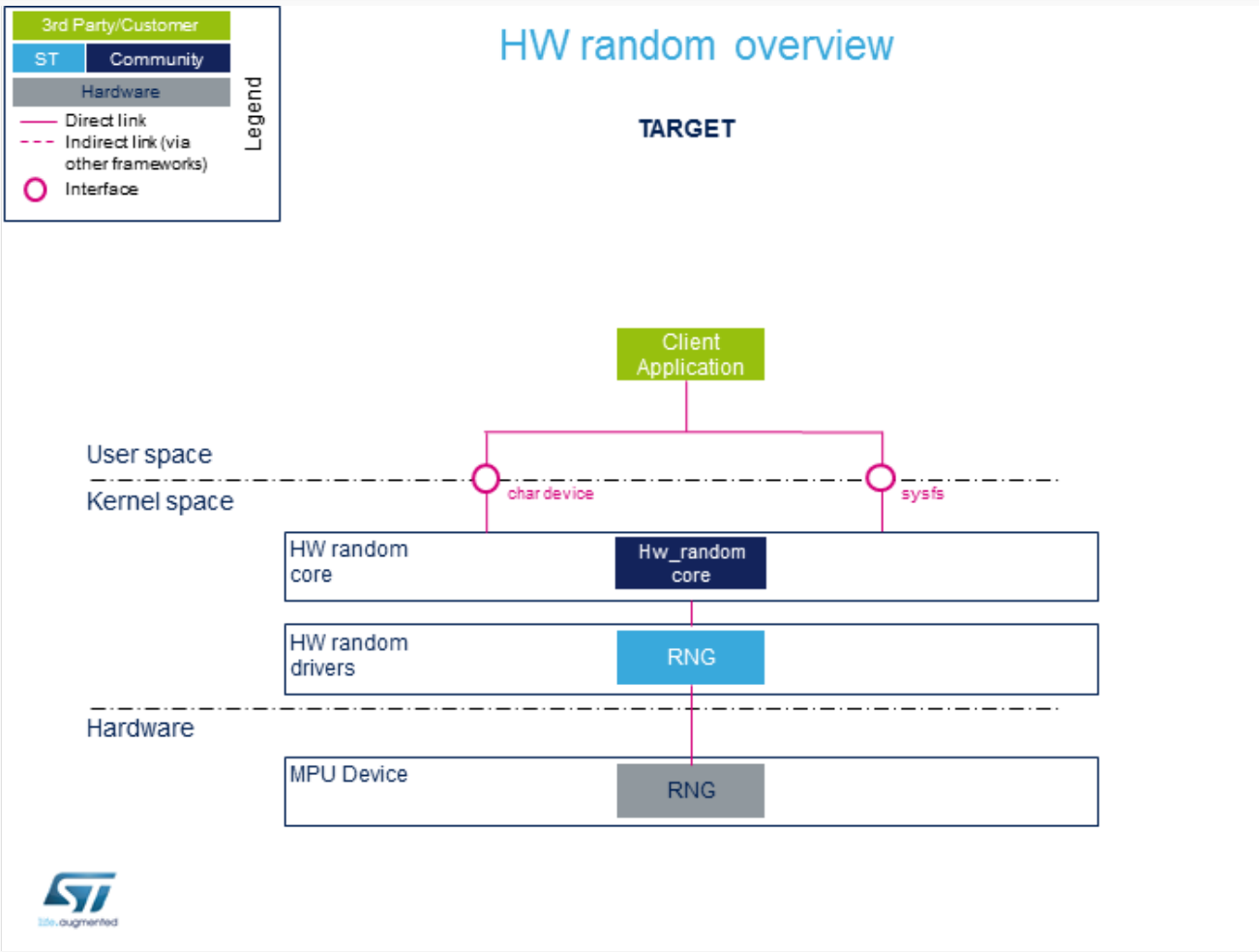
Contents

1 Framework purpose

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.



Hw Random

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.

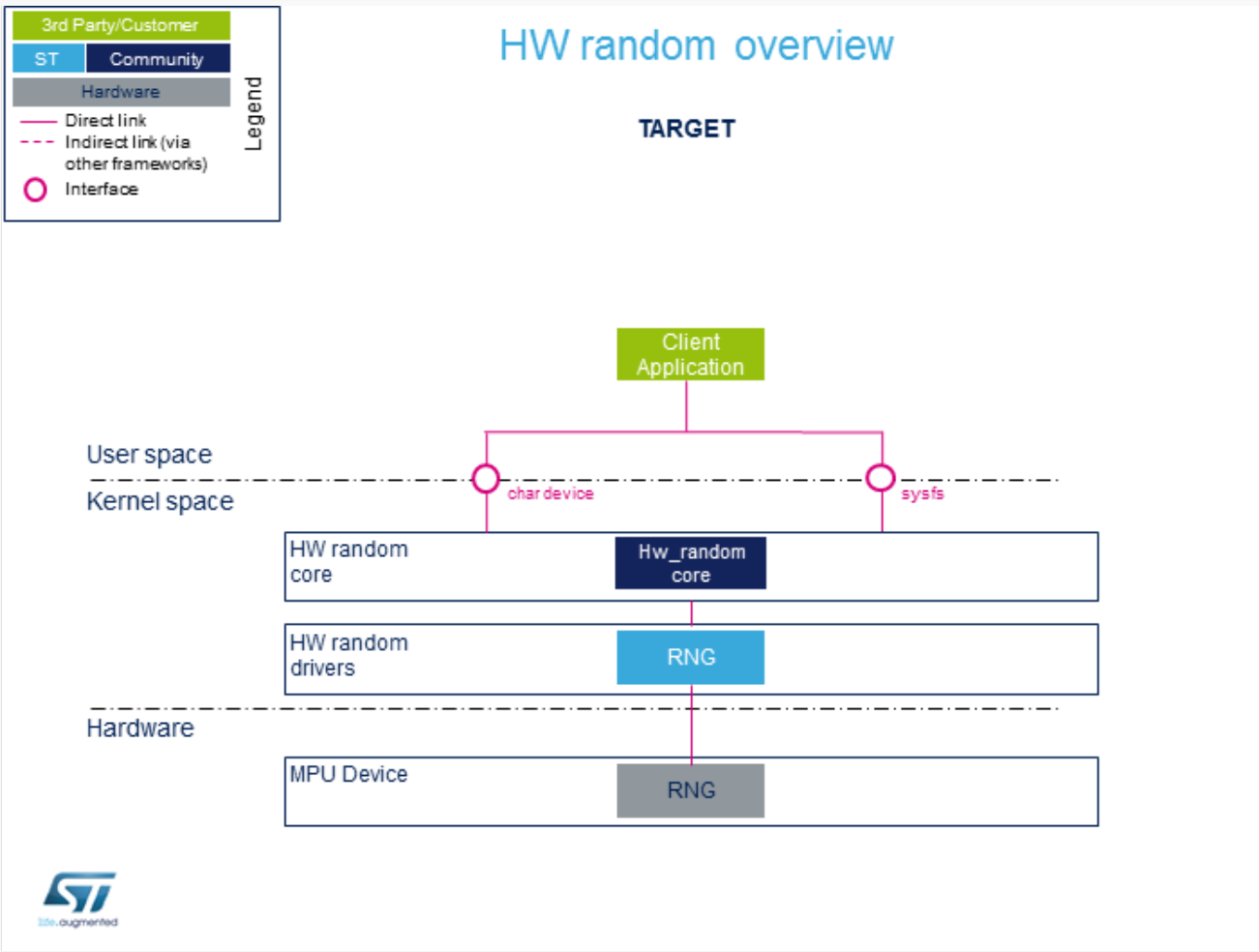
Contents

1 Framework purpose

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.



Hw Random

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.

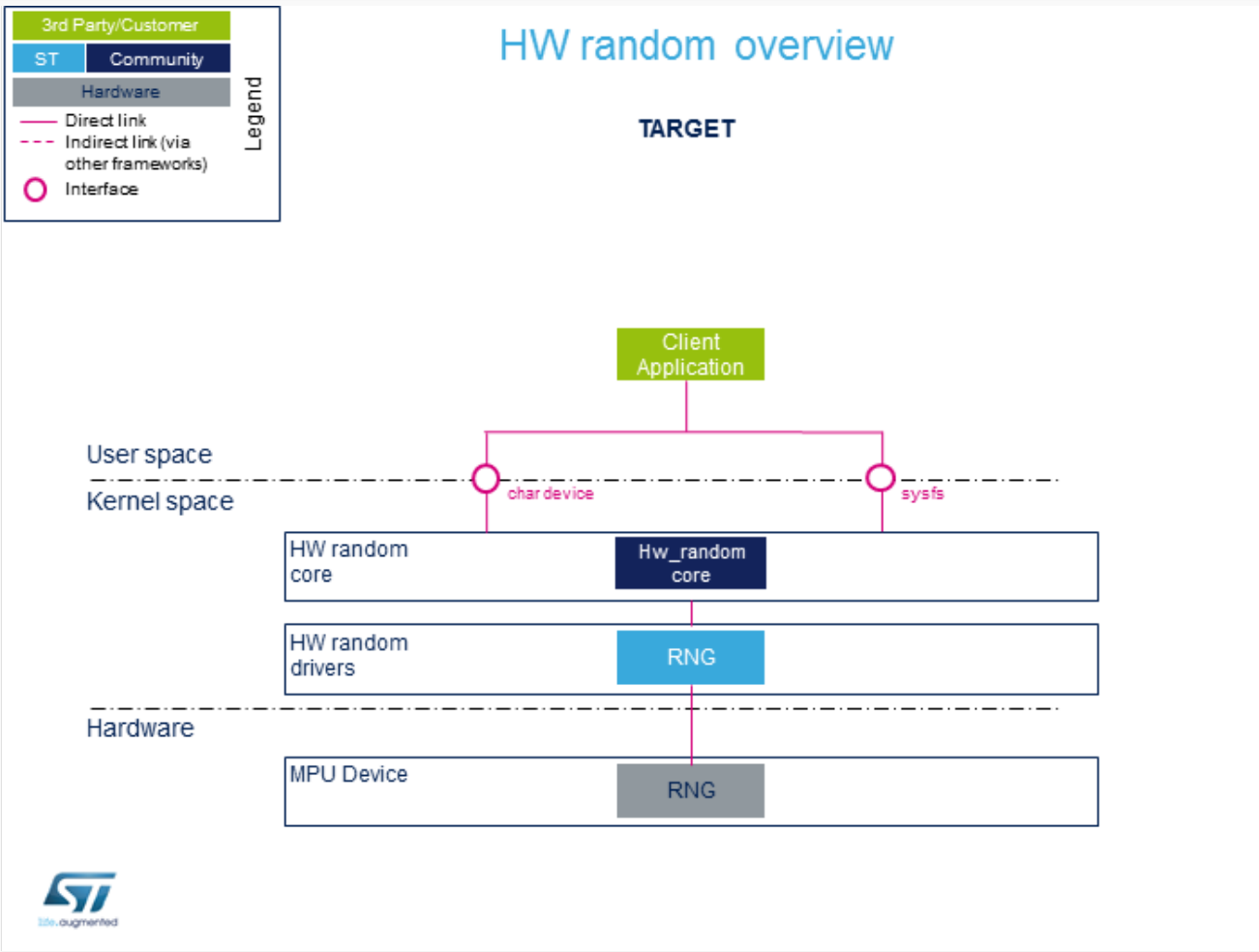
Contents

1 Framework purpose

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.



Hw Random

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