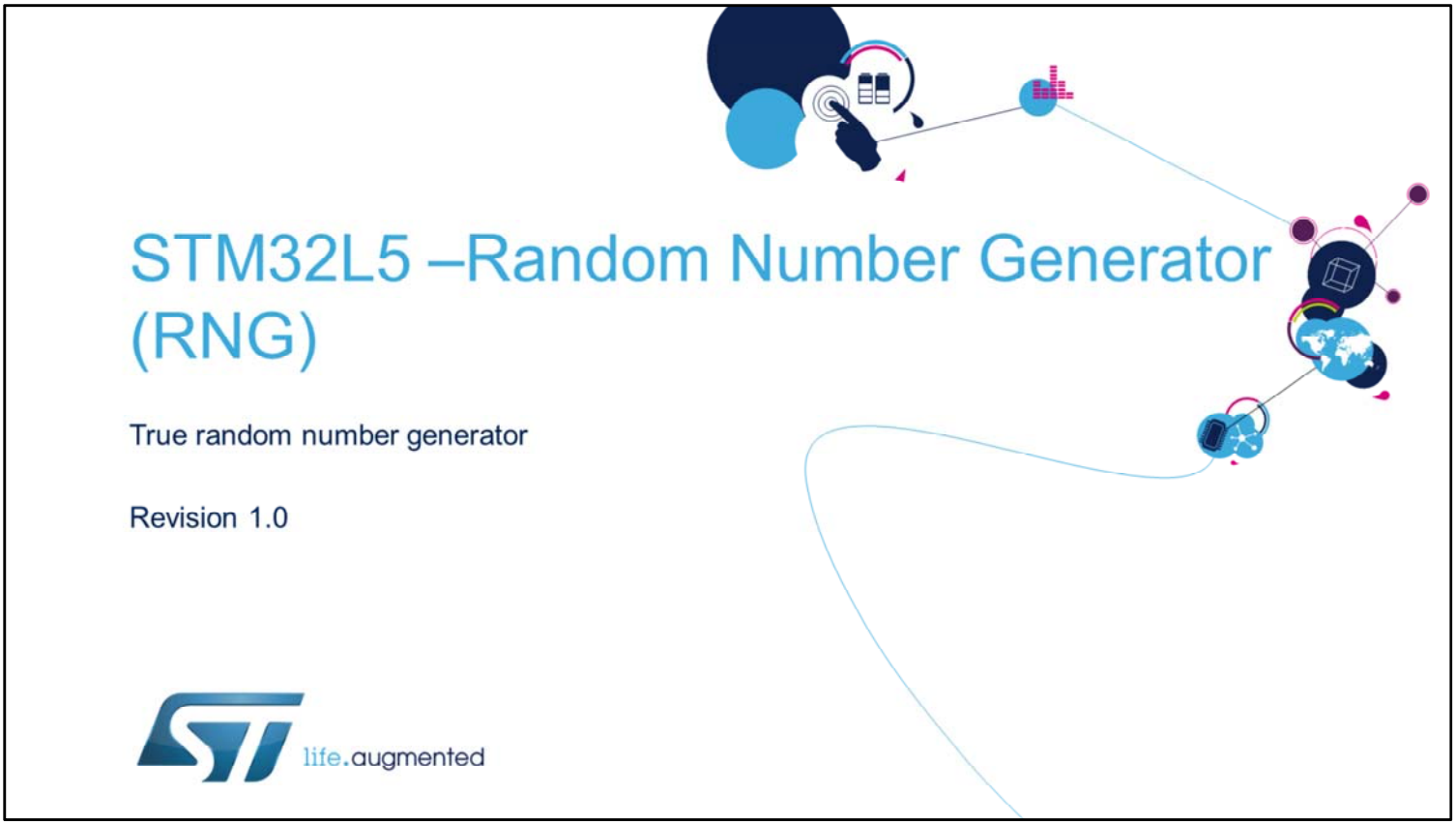
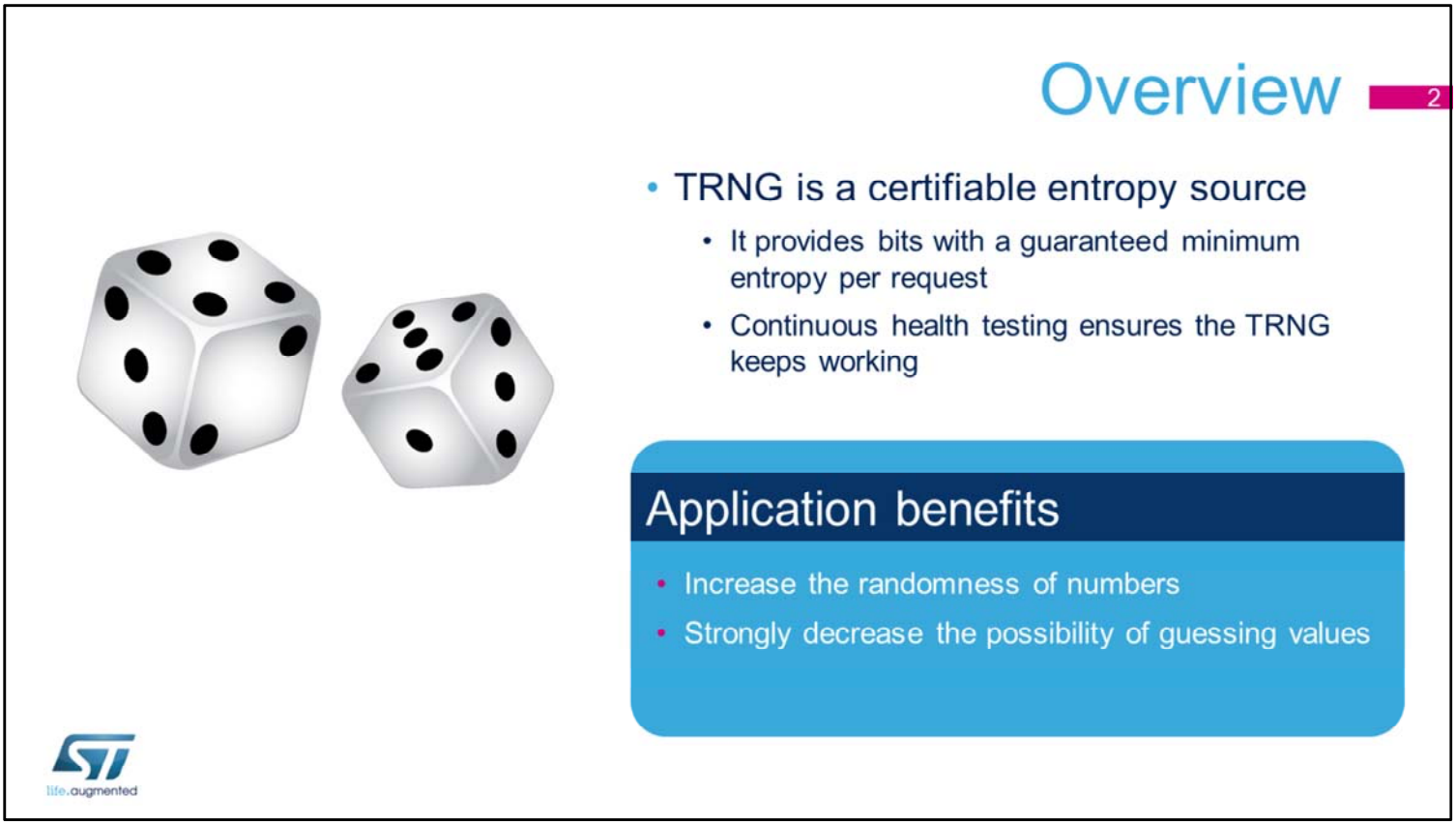


Hello, and welcome to this presentation of the STM32 Random Number Generator. The features of this peripheral, which is widely used to provide random numbers, will be covered in this presentation.



The random number generator (RNG) integrated inside STM32 products provides random numbers which are used when producing an unpredictable result is desirable. Applications can benefit from the RNG to increase the randomness of numbers or to decrease the possibility of guessing certain values.
It is certifiable NIST SP800-90B, with a guaranteed minimum entropy of 128 bits.

Key features

- 32-bit True Random Number Generator, certifiable NIST SP800-90B
  - In NIST compliant configuration, the time between two sets of four 32-bit data is either:
    - *412 AHB cycles if fAHB < f threshold*
    - *256 RNG cycles fAHB ≥ f threshold, with f threshold = 1.6 x fAHB, e.g. 77 MHz if fRNG= 48 MHz.*
  - When CLKDIV is different from zero, fRNG must take into account the internal divider ratio programmed in the RNG
  - Can be disabled to reduce power consumption (RNGEN=0 in RNG_CR).
- 3 flags can be triggered when:
  - DRDY: Valid random data is ready.
  - SECS: An abnormal sequence occurs on the seed, as specified by NIST SP800-90B specification.
  - *CECS:* f_{RNG} frequency is lower than f_{AHB} / 32 (this check can be disabled).
- 3 interrupts

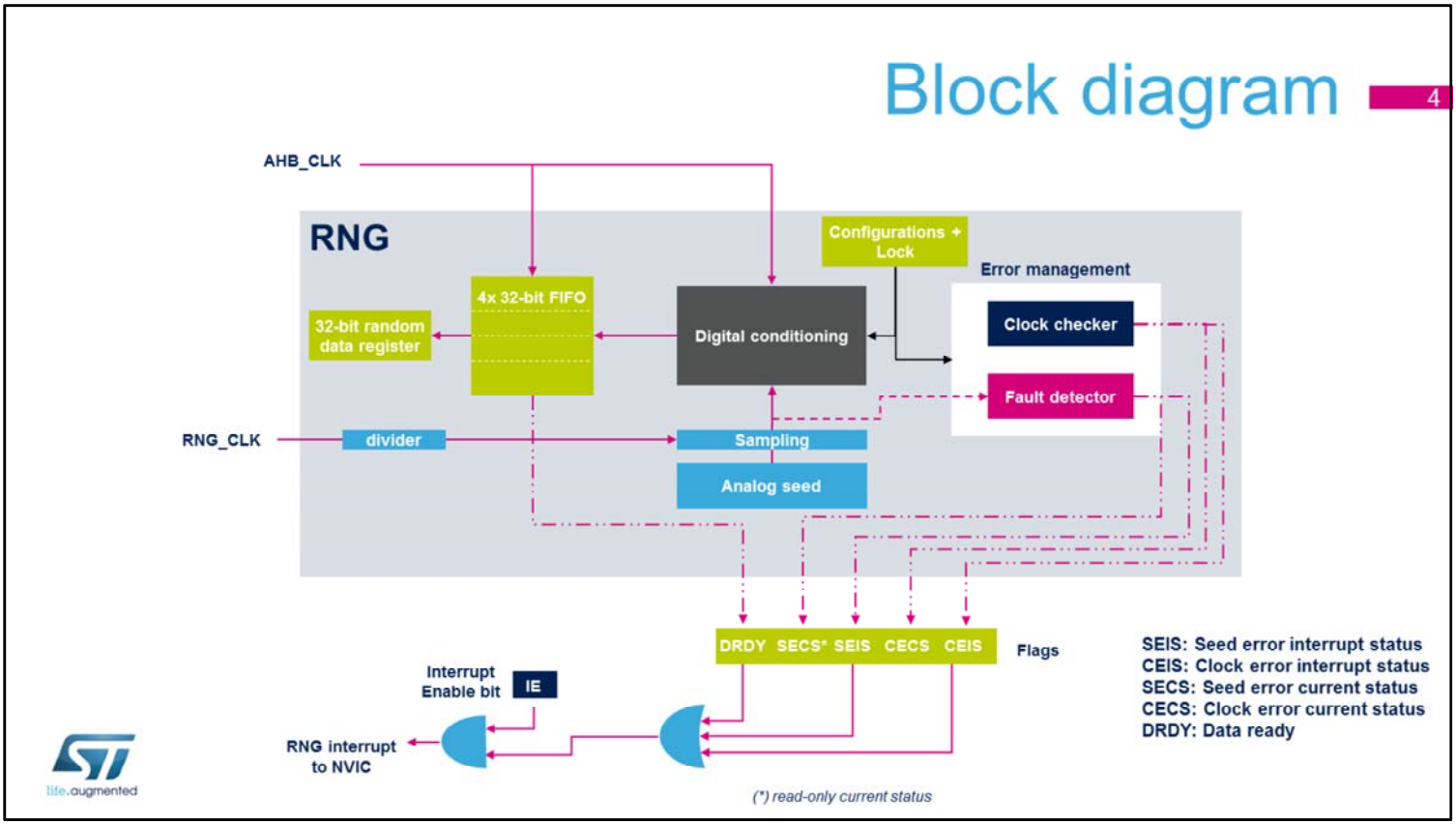
CEIS: to indicate a clock error	**SEIS**: to indicate a seed error	**DRDY**: to indicate that a valid random data is ready

life.augmented

The RNG peripheral is based on continuous analog noise that provides a random 32-bit value. It is certifiable NIST SP800-90B, with a guaranteed entropy of 128 bits.
When configured to pass NIST SP800-90B certification the time between two sets of four 32-bit data is either 412 AHB cycles if fAHB < f threshold, or 256 RNG cycles fAHB ≥ f threshold, with f threshold = 1.6 x fAHB (e.g. 77 MHz if fRNG= 48 MHz). When CLKDIV is different from zero, fRNG must take into account the internal divider ratio programmed in the RNG.

The Data Ready flag is set in the status register when a set of new random data is ready and validated. It must always be used.
The RNG automatically performs NIST SP800-90B compliant health tests on the the noise source (a Seed Error Current status flag is set in case of error).
A Clock Error Current status flag is set if the RNG clock is

less than HCLK clock divided by 32. This check can be disabled, especially when the RNG clock is initialized low for maximum entropy.
An interrupt source can also be enabled to indicate an abnormal seed sequence or frequency error.



This simplified block diagram of the RNG shows its basic functional and control modules. This architecture is compliant with NIST SP800-90B specification.
The random number generator is based on an analog circuit made of several ring oscillators whose outputs are sampled then XORed to generate the seeds that feed a digital conditioning block that is able to produce four 32-bit random numbers per round of computation.
The sampling of analog seeds is clocked by a dedicated RNG clock signal (+ internal divider) so that the quality of the random number is independent of the HCLK frequency. The contents of the conditioning block is transferred into the data register through a four-word FIFO. The Data Ready flag (DRDY) is triggered as soon as the FIFO is full, and is automatically reset when no more data can be read back from the RNG.
In parallel, an Error Management block verifies the correct seed behavior and the frequency of the RNG source clock.

Status bits are set and an interrupt is triggered if an abnormal sequence is detected in the seed or if the RNG frequency is too low.
The RNG frequency error check must be disabled if the RNG clock is fixed below AHB_CLK/ 32 (for example, for quality reasons).

TRNG block must be properly initialized with this sequence:

1) Write in the RNG_CR register the bit CONDRST=1 together with the correct RNG configuration.
2) Perform a second write to the RNG_CR register with the bit CONDRST=0, the interrupt enable bit IE=1 and the RNG enable bit RNGEN=1.

An interrupt is now generated when a random number is ready or when an error occurs.

Low-power modes

Mode	Description
Run	Active
Low power run	Disabled in RCC or in the RNG (RNGEN=0). Keeping RNG enabled remove latency before new random sample is available, because of the RNG initialization time
Sleep	
Low power sleep	
Stop 0 / Stop 1 / Stop 2	Disabled in RCC for lowest power consumption.
Standby	Powered-down. The peripheral must be reinitialized after exiting Standby mode.
Shutdown	

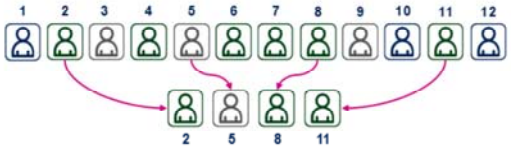
life.augmented

The true Random Number Generator is only active in Run mode. It can be kept enabled in Sleep and low power modes to avoid the latency at initialization time. It is disabled for the other low-power modes and is completely powered-down in Standby or Shutdown modes.

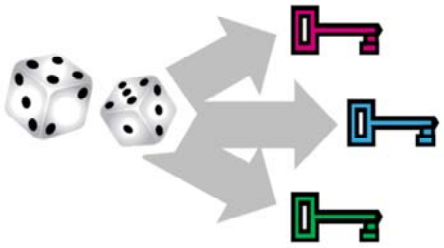
Application examples

- Lotteries, draws and gambling

- Random sampling



- Application to security
  - Generate keys, initialization vector, random padding, random challenge, digital signature…



life.augmented

The RNG can be used for a wide range of applications including cryptography, games, and statistical sampling. For example, all the security of cryptography algorithms are connected to the impossibility of guessing the key. So the key has to be a random number, otherwise the attacker can guess it.

Related peripherals

- Peripherals linked to the RNG
  - RCC (RNG clock control, RNG enable/reset)
  - Interrupts (RNG interrupt mapping)

life.augmented

This is a list of peripherals related to the random number generator. Please refer to these trainings for more information if needed.

References

- AN4230: STM32 microcontrollers random number generation validation using NIST statistical test suite.
  - Provides guidelines to run the suite of tests as specified in NIST SP800-90B
  - The NIST test suite was run on a selection of STM32 boards embedding RNG peripheral.

life.augmented

For more details, please refer to application note AN4230 about using the NIST statistical test suite to validate the random numbers generated by a selection of STM32 MCUs.