DOI: 10.35598/mcfpga.2023.016

Pseudo Random Value Generation in STM32 Cube

Oleksandr Vorgul
ORCID 0000-0002-7659-8796
dept. Microprocessor Technologies and Systems
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
oleksandr.vorgul@nure.ua

Iryna Svyd
ORCID 0000-0002-4635-6542
dept. Microprocessor Technologies and Systems
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
iryna.svyd@nure.ua

Oleg Zubkov
ORCID 0000-0002-8528-6540
dept. Microprocessor Technologies and Systems
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
oleh.zubkov@nure.ua

***Abstract*—This article is devoted to the description of the random number generator (RNG) in STM22 processors and study of the statistical properties of the values set being generated by the RNG. The analysis of the sequence of random numbers by statistical methods using the possibility of Matlab is given.**

Keywords—STM32, true random number, pseudo random number, random number generator (RNG), Matlab statistics toolbox

I. INTRODUCTION

Many STM32 processor families have a random number generation node on a board. According to the manufacturer datasheet [1], the random number is generated based on a physical sensor, that is, it is physically random. In any case, in the program interface there is no such thing as the initial value of the sequence (seed).

By the hardware cost, being compared to the processor part, RNG is a simple thing. But even for a simple node should be a reason why is it placed into the system [2-13].

II. DESCRIPTION OF THE RND HARDWARE PART

The block diagram from the manufacturer's documentation is as follows (Fig. 1).

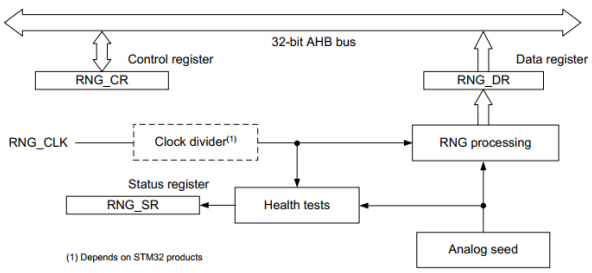

Fig. 1. STM32 true RNG block diagram.

Manufacturer sertified the RND by the NIST SP800-22rev1a test suite.

The NIST SP800-22rev1a statistical test suite is used to probe the quality of RNGs for cryptographic applications [2]. A comprehensive description of the suite is presented in the NIST document entitled A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications.

The NIST SP800-22rev1a statistical test suite "sts-2.1.1" is a software package developed by NIST that can be downloaded from the NIST web site (search for download the NIST Statistical Test Suite at csrc.nist.gov). The source code has been written in ANSI C. The NIST statistical test suite consists of 15 tests that verify the randomness of a binary sequence. These tests focus on various types of non-randomness that can exist in a sequence.

These test can be classified as follows:

- Frequency tests
  - Frequency (Monobit) test. To measure the distribution of 0's and 1's in a sequence and to check if the result is similar to the one expected for a truly random sequence.
  - Frequency test within a block To check whether the frequency of 1's in an Mbit block is approximately M/2, as expected from the theory of randomness.
  - Run tests To assess if the expected total number of runs of 1's and 0's of various lengths is as expected for a random sequence.
  - Test of the longest run of 1's in a block To examine the long runs of 1's in a sequence.
- Test of linearity
  - Binary matrix rank test To assess the distribution of the rank for 32 x 32 binary matrices.
  - Linear complexity test To determine the linear complexity of a finite sequence.
- Test of correlation (by means of Fourier transform)
  - Discrete Fourier transform (spectral) test To assess the spectral frequency of a bit string via the spectral test based on the discrete Fourier transform. It is sensitive to the periodicity in the sequence.
- Test of finding some special strings
  - Non-overlapping template matching test To assess the frequency of Mbit non-periodic patterns
  - Overlapping template matching test To assess the frequency of Mbit periodic templates
- Entropy tests
  - Maurer's "Universal Statistical" test To assess the compressibility of a binary sequence of L-bit blocks
  - Serial test To assess the distribution of all 2m Mbit blocks.

If results of the test are good then RNG values are random certified and life is beautiful.

III. MATLAB OFFERS

But nothing can stop us from generating good run from STM32F407VG microprocessor that is installed on STM32F4Discovery board to check it in the Matlab Statistics toolbox (Fig. 2).

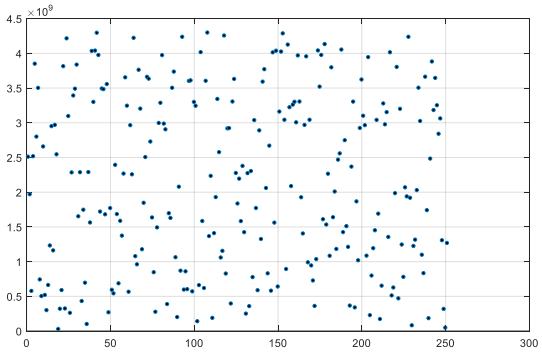

Fig. 2. Small set of random numbers.

The set that is screened on the Fig. 1 consists of 32 bit numbers, that is why the factor x10^9 on the vertical axes.

Distribution and disperse parameters for the set normalized to 1 are presented in the following Table 1.

TABLE I. STATISTICAL PARAMETERS

mean	median	range	standard deviation	geometric mean	harmonical mean
0.4980	0.0061	0.9939	0.2956	0.3721	0.1926

One can see that the mean is close to 0.5 and standard deviation is close to the theoretical value. These values are typical for the uniform distribution.

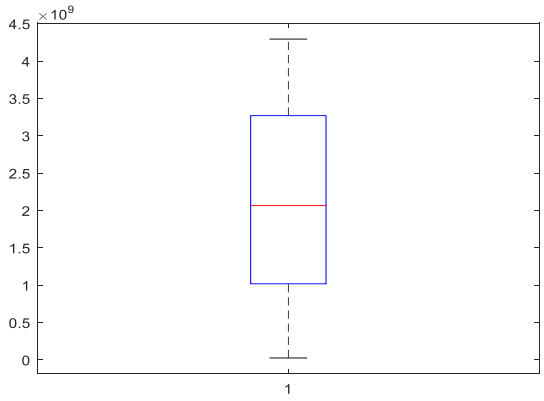

Fig. 3. Box and whiskers plot for non normalized case.

On the box plot one can see a complete coverage of the interval by the set members. The mean and 25% and 75% quartiles are symmetrically distributed as it should be for uniform distribution.

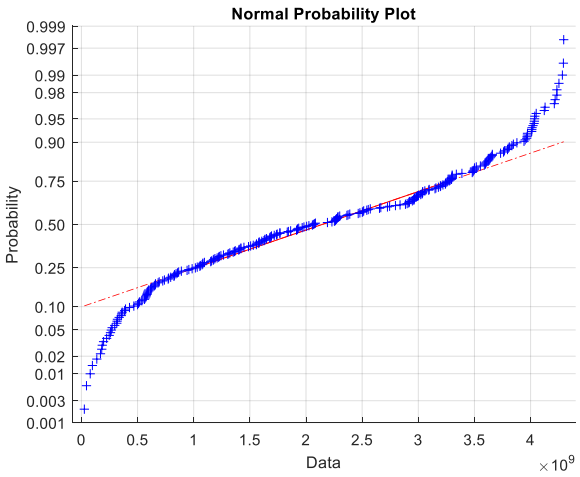

Fig. 4. Normal probability plot.

The Fig. 4 shows the discrepancy to the Gaussian distribution.

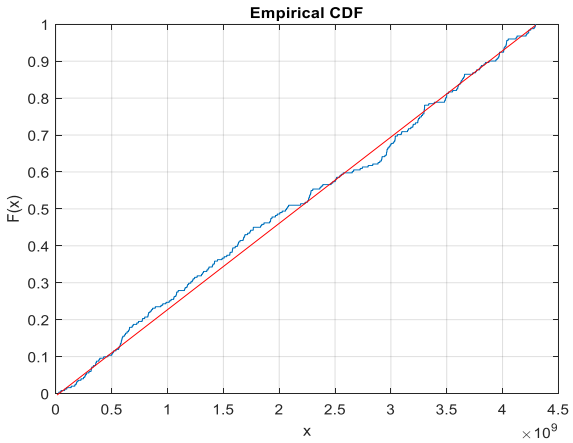

Fig. 5. Cumulative distribution plot.

Fig. 5 shows the empirical cumulative distribution function (cdf) for the data in the vector X. The empirical cdf *F(x)* is defined as the proportion of Xvalues less than or equal to *x*. On the Fig. 5 blue line corresponds to data points and red line is for the uniform distribution case.

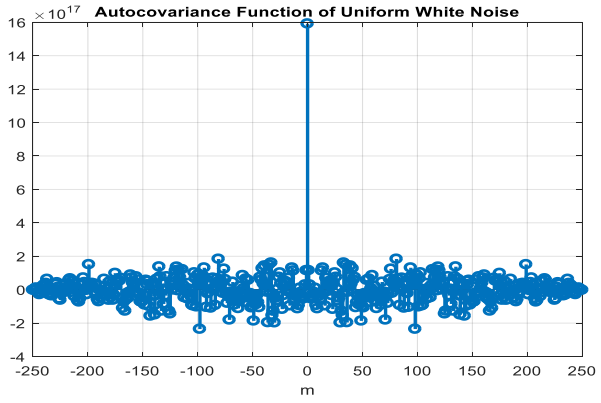

Fig. 6. Autocorrelation function.

Fig.6 is similar to the Delta function.

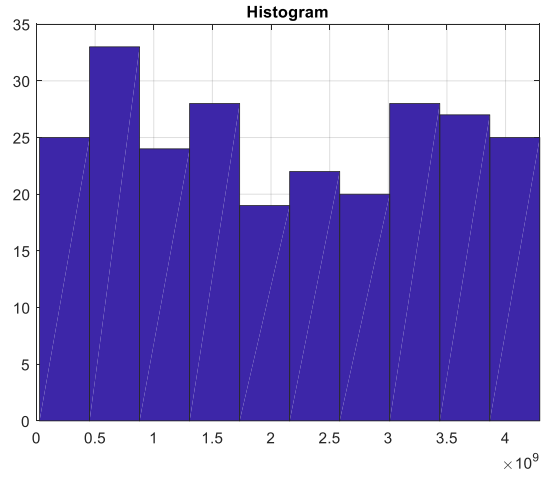

Fig. 7. Histogram of the random values set.

The histogram shown on the Fig. 7 is somewhat close to the uniform distribution. The difference from the uniform can be partially explained by small size of the set. The small size was chosen on purpose for the pattern on the Fig. 1 to be seen.

IV. Further Application

Among security problems, the official Random Number Generator document [3] provides application benefits of utilizing of RNG generated numbers:

- Increase the randomness of numbers,
- Strongly decrease the possibility of guessing values.

And all that was mentioned above seems to be promising, but why manufacturers insert such a device in the system? Among fields where hardware generated random value can be utilized are Monte Carlo something, white uniform noise generation and the base sequence generation for cryptographic purposes to name a few.

Cryptographic purposes seem to be resource consuming for a microcontroller and may require additional hardware support. But truly it must be linked to the good random number generator with a long length of sequence.

Using noise-like signal for modulation purposes is promising topic and requires separate research.

But to use source of physical noise for telecommunication purposes presumes two separated by long distance sites. And a problem of secret random sequence synchronization is appeared. And during exchange by the random sequence between two sites they may stop to be a secret.

V. Conclusion

Usually, to imitate the random value, the congruent method is used [4]. The algorithm is simple and fits the modeling condition: for debug purposes it is suitable to use the same sequence. It will not work for true random values unless one can store it in a good place. Disadvantage of the congruent method for cryptographic purposes is widely known – the sequence can be easily cracked.

More complex procedures of pseudo random value generation exists, but they either can be cracked or are required of serious hardware resources and can be cracked.

Random value generator on a board with microcontroller is a good thing, taking into account that it is not just a linear feedback shift register, but a linear feedback shift register that uses a physical random seed value from analog sensor. Besides, the entropy influence is regulated. In fact the more time interval in between two separate values the better theirs randomness.

References

[1] Random number generation validation using NIST statistical test suite for STM32 microcontrollers https://st.com/an4230

[2] JNIST statistical test suite. Random Bit Generation https://csrc.nist.gov/Projects/Random-Bit-Generation/Documentation-and-Software

[3] ISTM32WB-Security. Random Number Generator –RNG.pdf https://st.com

[4] David Knuth. The art of computer programming. Vol2 Seminumerical algorythms. Chart 3. Random Numbers - Addison Wesley.

[5] Програмування мікроконтролерів STM32 в середовищі STM32CubeIDE в прикладах і задачах: Навч. посіб. / О. В. Зубков, І. В. Свид, О. В. Воргуль, В. В. Семенець. Дніпро : ЛІРА ЛТД, 2022. 144 с.

[6] I. Svyd, V. Semenets, O. Vorgul, and I. Shevtsov, “Aspects of STEM education in the design of devices on microcontrollers and FPGAS,” Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs 2022, 2022. doi:10.35598/mcfpga.2022.018 .

[7] O. Zubkov, I. Svyd and O. Vorgul, "Features of the Digital Filters Implementation on STM32 Microcontrollers", 2021 III International Scientific and Practical Conference Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs, 2021. doi: 10.35598/mcfpga.2021.001.

[8] O. Zubkov, I. Svyd and O. Maltsev, "Features of the use of PID controllers when controlling evaporators", 2020 II International Scientific and Practical Conference Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs, 2020. doi: 10.35598/mcfpga.2020.001.

[9] O. Zubkov, I. Svyd, and O. Vorgul, “Features of the implementation of an over/under voltage relay on STM32 microcontrollers,” Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs 2022, 2022. doi:10.35598/mcfpga.2022.001.

[10] O. Vorgul, O. Zubkov, I. Svyd, and V. Semenets, “Teaching microcontrollers and FPGAS in quarantine from coronavirus: Challenges and prospects,” MC&FPGA-2020, 2020. doi:10.35598/mcfpga.2020.005.

[11] O. Vorgul, I. Svyd, and O. Zubkov, “Neuron networks design in Matlab and Vivado,” 2021 III International Scientific and Practical Conference Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs, 2021. doi:10.35598/mcfpga.2021.010.

[12] O. Vorgul, I. Svyd, V. Semenets, and O. Zubkov, “Enhancement of the laboratory workshop on FPGA: Opportunities and prospects,” Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs 2022, 2022. doi:10.35598/mcfpga.2022.010.

[13] I. Shevtsov, I. Svyd, V. Chumak, and A. Sierikov, “Practical aspects of software optimization for MCUS WITH RTOS,” Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs 2022, 2022. doi:10.35598/mcfpga.2022.012