Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 6 No. 2 (2022) 183 - 189 ISSN Media Electronic: 2580-0760

Gaussian Distributed Noise Generator Design Using MCU-STM32

M. Nanak Zakaria[1], Achmad S.[2], Ahmad Wilda Y[3], Lis Diana Mustafa[4]
[1,3,4] Electrical Engineering, Malang State Polytechnic, Jl. Soekarno-Hatta Malang, Indonesia
[2]Electrical Engineering, Gajayana University Malang, Jl. Mertojoyo Blok L Dinoyo Malang
[1]nanak_zach@polinema.ac.id, [2]ahmadseti@unigamalang.ac.id, [3]ahmadwildan@polinema.ac.id, [4]lis.diana@polinema.ac.id

Abstract

The random noise signal is widely used as a test signal to identify a physical or biological system. In particular, the Gaussian distributed white noise signal (Gaussian White Noise) is popularly used to simulate environmental noise in telecommunications system testing, input noise in testing ADC (Analog to Digital Converter) devices as well as testing other digital systems. Random noise signal generation can be done using resistors or diodes. The weakness of the noise generator system using physical components is the statistical distribution. An alternative solution is to use a Pseudo-Random System that can be adjusted for distribution and other statistical parameters. In this study, the implementation of the Gaussian distributed pseudo noise generation algorithm based on the Enhanced Box-Muller method is described. Prototype of noise generation system using a minimum system board based on Cortex Microcontroller or MCU-STM32F4. From the test results, it was found that the Enhanced Box-Muller (E Box-Muller) method can be applied to the MCU-STM32F4 efficiently, producing signal noise with Gaussian distribution. The resulting noise signal has an amplitude of ±1Volt, is Gaussian distributed and has a relatively wide frequency spectrum. The noise signal can be used as a jamming device in a certain frequency band using an Analog modulator.

Keywords: Noise Generator, Gaussian Distribution, Enhanced Box-Muller, MCU-STM32F4

1. Introduction

In industrial environment, signal noise is the main factor that causes errors in signal processing. Errors in making decisions by the processing system generally occur because the noise signal damages the main signal to be processed. The consequence of mixing the main signal with the noise signal is that an unexpected response is generated by the system and this will lead to other undesirable consequences, such as damage to the resulting product [1][2]. Based on the provisions in [1][2], it can be concluded that the ability of the entire system to be able to process information correctly and be able to overcome interference from noise is the main requirement to obtain the good performance of the system.

In the development of signal processing systems, both analog processing systems and digital processing systems, testing the performance of the system against noise is carried out by testing the response of the system to the composite input signal, namely the input signal that has been mixed with the noise signal [1][2] . The noise signal is attempted to be as similar as possible to the noise signal that appears in actual environmental conditions. This is something important considering that disk reader systems, telemetry systems and radio-based control systems always pick up a signal mixed with noise for further processing. A processing system is called a high-performance system if the system can suppress the noise collected along with the input signal. To test the ability of these systems against noise interference, especially testing on a laboratory scale, a device that can generate noise signals is needed as well as actual environmental noise [2]. This kind of environmental noise is known as white noise.

The problem faced in the design of a noise generator system is how to generate a random signal that has a certain distribution (uniform or gaussian) efficiently. From the literature search, the authors get a lot of literature that writes about methods of generating random signals either purely (pure random) or artificial (pseudo random) [3].

Gaussian distributed noise signal can be generated artificially (pseudo random) using a mathematical algorithm or purely (pure random). The noise generation method is purely done by amplifying the electron activity signal in a diode. This pure noise signal

Accepted: 05-12-2021 | Received in revised: 25-02-2022 | Published: 20-04-2022

has no repetition period or can also be referred to as a random signal with an infinite period.

The artificial noise generation method generally uses a mathematical algorithm (to generate a series of random numbers) which is then converted into an analog signal using a DAC (Digital to Analog Converter). The main weakness of the noise generation method using this mathematical algorithm is the appearance of periodization in the resulting signal [3][4][5]. There is a lot of literature that focuses its research on developing algorithms to generate a series of random numbers close to the characteristics of pure random numbers. One method that can be used to make the pseudo random sequence close to pure random is to apply a large random number repetition period. The consequence of this method is the inclusion of large-value coefficients in the mathematical system used.

One method to generate a Gaussian distributed noise signal is to use the Box Muller method [6][7]. In this paper, the mathematical formulation of the Box-Muller method is implemented on an FPGA chip for further processing using an analog modulator [6]. The modulator will place the resulting noise signal in the desired band. The Box-Muller method was later improved by Adnane et al. in [8] in the "tail" section of his Gaussian response.

Based on papers [6][8], this paper will describe the implementation of the Enhanced Box-Muller method on the MCU-STM32 to generate a White Gaussian Noise signal. The use of MCU (Micro Controller Unit) as a processing unit in this study was chosen because of its flexibility in the implementation and control process.

2. Research Methods

2.1 Noise Generating System

In testing the performance of signal processing devices, a measurable and easily reproducible Noise signal is absolutely necessary. This kind of noise is also known as pseudo noise because in reality the noise signal repeats itself for a certain period. This condition is different if the Noise signal is not repeated or has no period. This kind of noise is called pure noise which can be generated from the activity of electrons on a transistor strand, on a diode or it can also be done by sampling the noise signal on an ADC (Analog to Digital Converter) input pin.

To create a Pseudo Noise generator system, a random number generator system and a DAC (Digital to Analog Converter) device are required which are arranged as shown in Figure 1.

From Figure 1 it can be seen that the noise signal is generated by converting random digital data generated by the Digital Random Generator (DRG) into an analog signal using a DAC device. This means that if the DRG generates a Gaussian distributed random number, the noise signal it produces will also be Gaussian distributed as well.

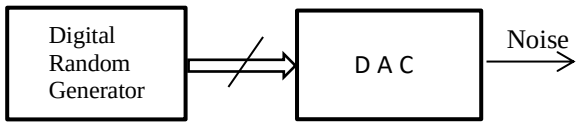

Figure 1. Pseudo Noise Generator Diagram

2.2 Box-Muller method

From Figure 1 it can be seen that the DRG unit is the most important unit in a noise generation system. Regarding portable and simple algorithmic requirements, many studies have been carried out so far to obtain an efficient random number generation algorithm. The algorithm is then realized in the form of hardware [4] and in the form of software such as the Central limit method, the Ziggurat method, the Inversion method, the Wallace method and the Box-Muller method.

The Box-Muller method is a random number generation method that has a simple, portable and easy algorithm to implement at the hardware level (FPGA and Microcontroller). The Box-Muller method is a Gaussian distributed random number generation method that converts 2 (two) Uniform distributed random numbers into 2 Gaussian distributed random numbers that are perpendicular to each other (Quadrature) [6][7][8].

In general, the Box-Muller transform is written as follows: if there are 2 (two) independent Gaussian distributed random numbers called X and Y, then probability density function (PDF) is expressed as:

$$f(x,y) = P(x).P(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} . \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}$$
$$= \frac{1}{2\pi} e^{-(x^2+y^2)/2} \quad (1)$$

Because PDF in equation (1) is distributed radially, then the equation can be expressed in polar form as (R, Θ). The value of angle Θ has a range of $0 \leq \Theta \leq 2\pi$, while X and Y are defined as $X = Rcos\,\Theta$ and $Y = R \sin\Theta$. The parameter Θ will be uniformly distributed from 0 to 2π so that can be expressed as $\Theta = 2\pi U_1$.

The parameter R itself is expressed as $R = \sqrt{-2 \ln U_2}$. Referring to the previous definitions, the Box-Muller method can be written as Equation (2) below:

$$\begin{cases} X = \sqrt{-2ln(U1)}\ \cos(2\pi U2) \\ Y = \sqrt{-2ln(U1)}\ \sin(2\pi U2) \end{cases} \quad (2)$$

U1 and U2 are a pair of Uniform distributed random numbers to be converted, while X and Y are a pair of

DOI: https://doi.org/10.29207/resti.v6i2.3684

Gaussians distributed random numbers converted from U1 and U2.

If equation (2) is realized in the form of a system block diagram, then Equation (2) can be described as follows:

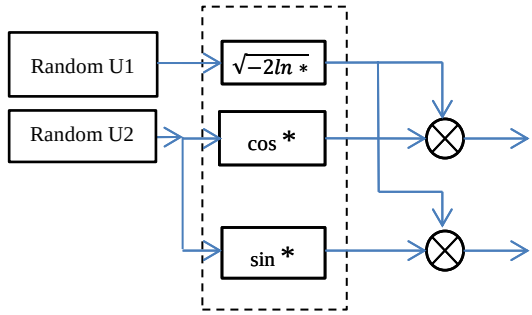

Figure 2. Block diagram of the Box-Muller method

This relatively simple Box-Muller method was further developed by Brent [9]. Brent modified the Box-Muller formula by converting Trigonometric terms into Non-Trigonometric terms. This technique is used to simplify the calculation process if the Box-Muller method will be applied to a simple digital processing system. This Brent formula is also known as the polar form of the Box-Muller formula, and is written as follows:

$$\begin{cases} X = v_1 \sqrt{\dfrac{-2\,ln\,(v_1^2+v_2^2)}{(v_1^2+v_2^2)}} \\ Y = v_2 \sqrt{\dfrac{-2\,ln\,(v_1^2+v_2^2)}{(v_1^2+v_2^2)}} \end{cases} \quad (3)$$

2.3 Enhanced Box Muller Method (E Box-Muller)

The enhanced Box-Muller method or also known as the Enhanced Box-Muller is a method proposed by Adnane Addaim et. all. to improve upon the classic Box-Muller method on the "tail" side of the Gaussian distribution. The weakness of the classic Box-Muller method is that it is not suitable for producing high values of **σ**. In the Box-Muller method, the value of **σ** will reach its maximum if 2 (two) converted random numbers (U1 and U2) together reach a value of 0. Because U1 and U2 are two independent numbers, this condition is very difficult to achieve. So it is proposed to use a Uniform distributed random number, say **U**, which is applied to the classical Box-Muller formula. The improved Box-Muller formula can be stated as follows:

$$\begin{cases} X = \sqrt{-2ln(\boldsymbol{U})}\,\cos(2\pi n\boldsymbol{U}) \\ Y = \sqrt{-2ln(\boldsymbol{U})}\,\sin(2\pi n\boldsymbol{U}) \end{cases} \quad (4)$$

If equation (4) is compared with equation (2), it can be seen that Adnan et al. in [8] developed the Box-Muller formula in equation (2) by adding the n term to the Trigonometric term and using only one Uniform distributed random number (**U**).

2.4 Lehmer's Method for Generating Uniform Random Numbers

From Figure 1, it can be seen that to generate Gaussian distributed random numbers using the Box-Muller method, a uniformly distributed random number is required. To generate random numbers with uniform distribution, the Lehmer method [3][10] can be used. Mathematically, the Lehmer method for generating random numbers can be written as:

$$X_{n+1} = (\,aX_n)\ mod\ m \quad (5)$$

The Lehmer method is also known as the PMMLCG (Prime Modulus Multiplicative Linear Congruential Generator) method. Equation (5) above requires an initial value known as *the seed*. Although this method is relatively simple, it requires more attention to choose the values of **a** and **m** so that the resulting series of random numbers does not repeat itself as illustrated in [11][12]. Determination of the values of **a** and **m** is generally based on the ability of the machine and the desired iteration period.

If the digital machine used is a 16-bit machine, then selecting the value of **m** of 2^{16} will make a series of random numbers that are generated repeated every 2^{16} - 1. This condition can occur if **a** is the primitive root of **m**[1]. One option value a is 25173 and is used in Turbo Pascal software which is fully defined as:

$$X_{n+1} = (25173X_n + 13849)\ mod\ 2^{16} \quad (6)$$

X_{n+1} is a Uniform distributed random number, X_n is the initial value (seed) while the *mod* operation is the remainder division operation.

2.5 Noise Generation System using MCU-STM32

The Gaussian Distributed Noise generator system can be diagrammatically described as follows:

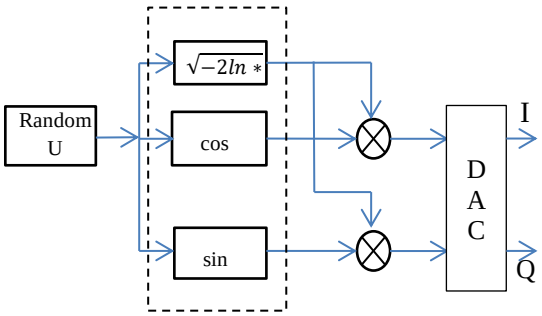

Figure 3. Noise Generator Model E Box-Muller

From Figure 3 it can be seen that the Noise Generator system designed using the E Box-Muller Method has 4 (four) basic parts, namely: a Uniform Distributed Random Generator Unit, an E Box-Muller Converter Unit, a Multiplier Unit (Multiplier) and a DAC (Digital to Analog) unit. converters). The E Box-Muller

DOI: https://doi.org/10.29207/resti.v6i2.3684

converter unit consists of a weighting unit, a sinusoidal unit and a cosinusoidal unit. Random Number Generator is a unit that generates Uniform distributed random numbers. This system is triggered by a clock unit that performs the sampling process with the desired sampling rate. Every time the sampling signal appears, this unit will generate a uniformly distributed random number with the interval [0,1]. The random number is then used as input for the E Box-Muller converter (Weighting, Sine and Cosine) so that a Gaussian distributed random number is generated. The algorithm of the Noise generator system as shown in Figure 3 can be stated as follows:

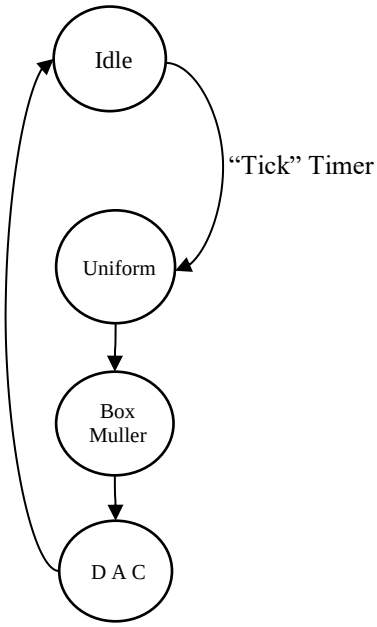

Figure 4. Flow state of the Noise generating system.

From Figure 4, it can be seen that the state change process that occurs in the Noise generator system, where the system will enter the Idle state if there is no "tick" signal from the Timer. If the signal appears, there will be a transition from the Idle state to the Uniform state. In this state, a Uniform distributed random number will be generated. The next transition is the E Box-Muller Conversion which will convert the Uniform random number into a Gaussian random number. The transition then proceeds to the DAC state where the Gaussian random number is converted into an Analog signal. The process continues to the Idle state to wait for the "tick" signal from the Timer in the next cycle.

2.6 Hardware System

Departing from the concept that the generator system must be made as compact as possible, the use of a Microcontroller (MCU) from the STM32 family is the right choice. Especially the STM32F407 µC chip which has features as a DSP (Digital Signal Processing) chip. One of these features is the availability of 2 (two) 12-bit DAC devices that can be programmed freely in addition to the system clock which reaches a value of 200MHz.

In the design of the Noise Generating System, all the units depicted in Figure 3 and Figure 4 will be implemented in the form of software on the STM32 Microcontroller. There are 2 (two) main units of the STM32 Microcontroller that play a role in the design of this Noise Generator, namely: DAC unit and Timer.

The block diagram of the DAC unit in the STM32 Microcontroller can be shown as Figure 3 below [13]:

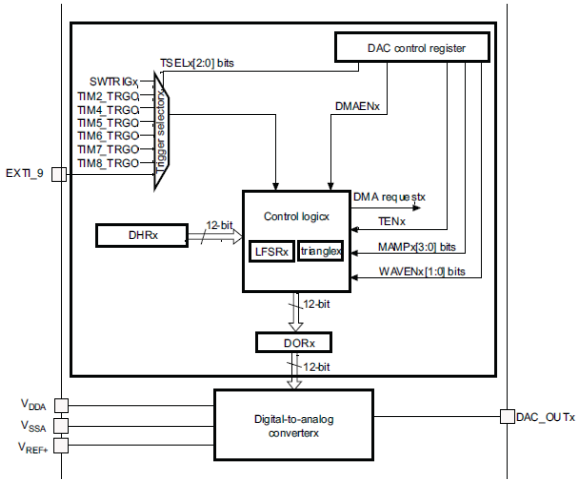

Figure 5. DAC unit structure in STM32

From Figure 5 it can be seen that the DAC in STM32 can be accessed manually or using a DMA (Direct Memory Access) mechanism. The manual mechanism is carried out by filling the buffer directly from each DAC device (DHRx register). While the DMA mechanism is carried out using Timer tools which periodically fills the buffer of the DAC without having to involve the CPU directly. In the design of this Noise Generation System, the DAC is accessed manually, where every 0.5ms the DAC buffer is filled by the E Box-Muller unit.

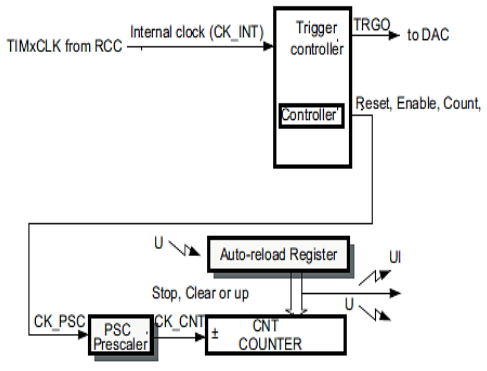

Figure 6. Timer structure on STM32

The second unit used in this design is the Timer unit. The Timer unit is used to tell the data processing unit (CPU) to perform the Uniform random data generation

DOI: https://doi.org/10.29207/resti.v6i2.3684

process and perform the calculation process based on the E Box-Muller (Enhanced Box-Muller) algorithm. The simple structure of the Timer in STM32 can be seen in Figure 6 [13].

In general, Figure 6 informs that the timer is made up of the internal clock (CK_CNT), counter (CNT), register reload and interrupt mechanism. Each clock signal appears, then the Counter will count up. If the value in the Counter reaches its maximum value, an Interrupt signal will appear which forces the Microcontroller to enter the routine implementation of the Timer service. The timer in this design is used as a "tick" signaling device that forces the CPU to carry out routine Uniform random number generation, conversion of uniform random numbers to Gaussian random numbers and filling of calculated data into each DAC buffer as shown in Figure 4.

The flow diagram of the Noise Generating System can be described as figure 7.

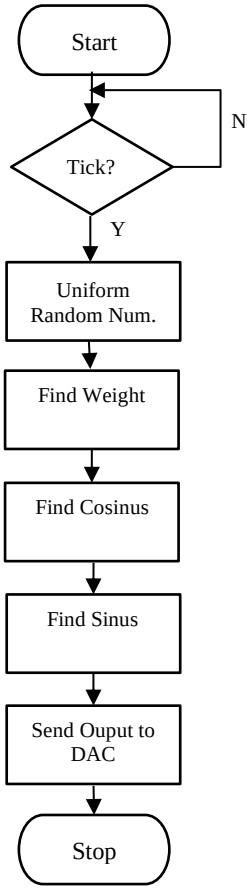

Figure 7. Flowchart of Noise Generator System

From figure 7, it can be seen that the time flag checking process begins to give orders to the CPU to perform the process of generating Uniform Distributed Random Values, then calculating the Weights, Cosines and Sines. The three values are then multiplied as the E Box-Muller formula to produce 2 random values. The two random values are then placed in each DAC buffer to form an orthogonal analog signal.

3. Results and Discussions

To get the performance value of the designed Noise Generator System, a Digital Oscilloscope measuring device can be used which has FFT (Fast Fourier Transform) mathematical facilities. This facility is used to get a spectrum picture of the resulting noise signal.

Testing using a Digital Oscilloscope at the DAC output or the I-Q point in Figure 3 produces the following Oscillogram:

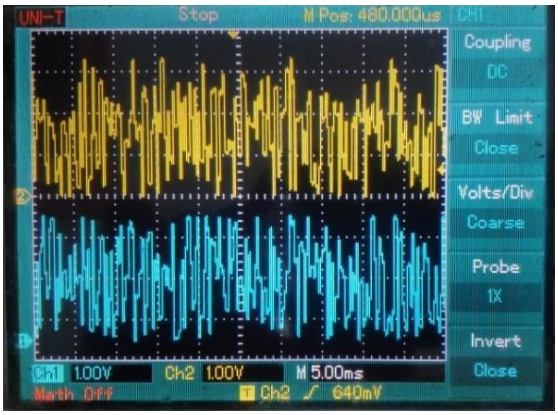

Figure 8. Oscillograms of I and Q. signals

From Figure 8 it can be seen that the noise signal (signal I and signal Q) generated by the system oscillates at a voltage value of ± 1V (2Vpp). This stress value is in accordance with the results of theoretical calculations in Equation (4).

To obtain information about the distribution of random numbers generated by the E-Box-Muller unit, the random number values generated by STM32 in branch I (Figure 3) were 5000 samples for 2 (two) variations in the value of **n** (i.e. n = 10 and n = 1000). The recording results are then processed to obtain information about the distribution of the random number values generated.

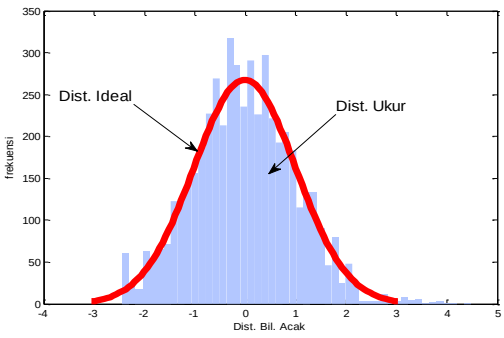

Figure 9a. Branch distribution I with n = 10 for 5000 samples

DOI: https://doi.org/10.29207/resti.v6i2.3684

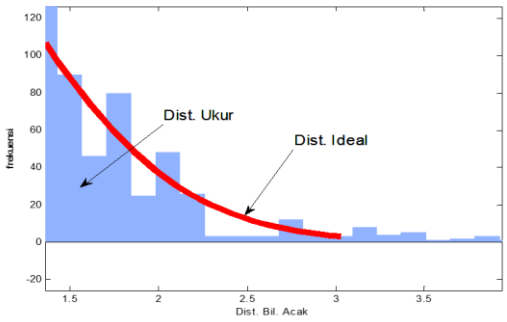

Figure 9b. Distribution at the tail end

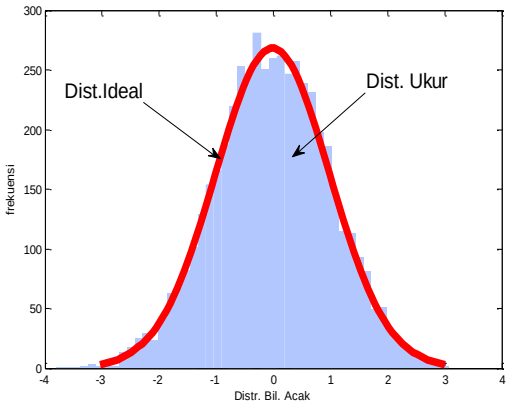

Figure 10a. Distribution of branch I with n = 1000 for 5000 samples

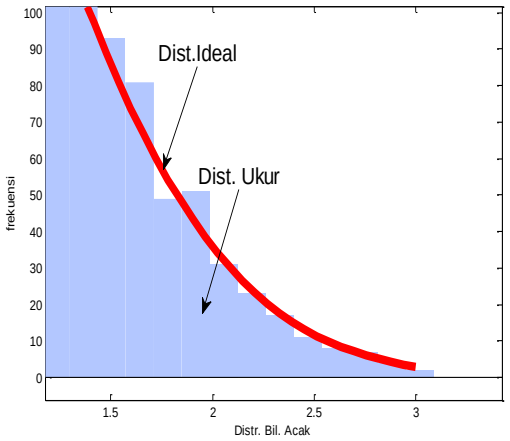

Figure 10b. Distribution at the tail end

Figure 9a and Figure 10a show the distribution of random numbers generated by the Enhanced Box Muller unit for n = 10 and n = 1000. From the two figures, two conclusions can be drawn, namely: that the Enhanced Box Muller unit designed to produce numbers normally distributed or Gaussian distribution. The second conclusion is that the value of n in equation 4 affects the "shape" or envelope shape of the resulting distribution, where for n = 1000 shapes the measured distribution approaches the ideal Gaussian distribution while for n = 10 the distribution of random numbers shows a distorted shape compared to the distribution. Gaussian ideal. The previous conditions will also apply to the tail end of the distribution as in Figures 9b and 10b.

To find out the distribution of the frequency or spectrum generated from the designed device, a spectrum test was conducted using a spectrum analyzer. The test is carried out by measuring the output in one of the branches, namely branch I. The measurement results in that branch are illustrated in Figure 11.

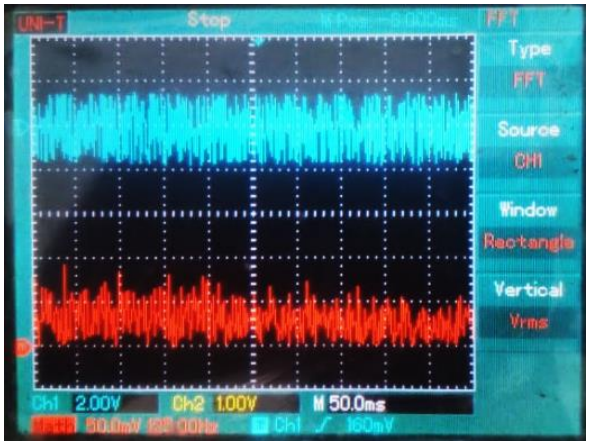

Figure 11. Noise and Noise Spectrum at branch I

Figure 11 above shows the noise signal pattern generated by the Enhanced Box-Muller after being passed to a 12-bit DAC unit and the resulting frequency spectrum pattern. The resulting noise signal seems to swing from -1 Volt to +1 Volt or 2Vpp.

The spectrum analysis of the generated noise signal can be seen in the second part of Figure 11 (Noise Spectrum). In the image it can be seen that the noise spectrum stretches up to 12KHz. The stretch of the frequency spectrum up to 12KHz with relatively the same amplitude indicates that the generated noise signal meets the requirements to be called a white noise signal.

If the noise is modulated using a carrier signal with a frequency of 200kHz, the oscillogram is obtained as follows:

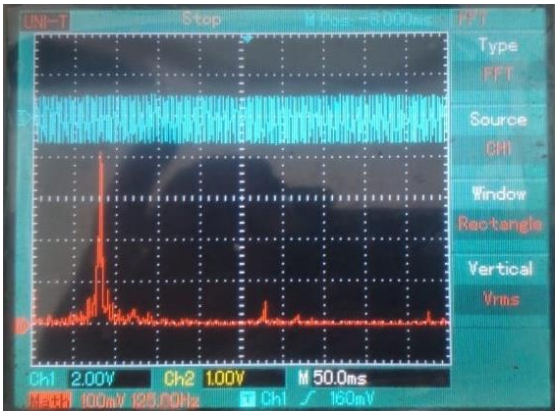

Figure 12. Noise signal modulated using fc = 200KHz

DOI: https://doi.org/10.29207/resti.v6i2.3684

From Figure 12 it can be seen that if the noise signal is modulated (in this study, fc = 200KHz was used), then the noise spectrum will be centered on the carrier frequency, namely fc = 200KHz. This indicates that by means of an analog modulator, the noise spectrum can be centered on the desired frequency band.

4. Conclusion

The E-Box Muller method that has been described on the theoretical basis can be implemented on the MCU-STM32F4 hardware directly and generates Gaussian distributed random numbers as described in the theory.

The resulting Noise signal has a voltage that stretches from -1 to +1 Volts and produces a frequency spectrum that stretches up to 12KHz with an average spectrum amplitude of 50mV. This proves that the noise signal meets the requirements to be referred to as White Noise.

If the noise signal is modulated using an analog modulator, then the frequency spectrum of the noise signal can be focused on a certain frequency band. This means that the designed noise generation system can be developed as a jammer device or a signal processing device performance test system.

Acknowledgment

The author would like to thank the Research and Community Service Unit (P2M) of the Malang State Polytechnic which has provided funding through the 2021 DIPA Research and Community Service program.

Reference

[1] "Reducing Signal Noise in Practice @ www.predig.com", [Online]. Available: www.predig.com/whitepaper/Reducing Signal Noise in Practice, February 2016

[2] P. Coanda, M. Avram, D Comeaga, "A hands-on approach to demonstrating active noise canceling", IOP Conf. Series: Materials Science and Engineering 997 (2020) 012040, 2020, doi:10.1088/1757-899X/997/1/012040

[3] Pierre L'Ecuyer , "History Of Uniform Random Number Generation", Proceedings of the 2017 Winter Simulation Conference pp. 202-230, 2017

[4] Edward D. Lipson, Keneth Foster and Michael p. Wals, "A Versatile Pseudo-Random Noise Generator", IEEE Transaction on Instrumentation and Measurement, Vol.25, No. 2, 1976

[5] Stephen K., Keith W. Miller, "Random Number Generators: Good Ones are Hard to Find", Communication of The ACM Vol. 31, 1988

[6] Liu Bodong, "The Design and Implementation of BroadBand White Noise Generator based on AD9957", IEEE International Conference on Electronic Information and Communication Technology, 2016

[7] Aleksei F. Deon and 2Yulian A. Menyaev , "Twister Generator of Random Normal Numbers by Box-Muller Model", Journal of Computer Science 2020, Vol.16 No. 1, 2020, DOI: 10.3844/jcssp.2020.1.13

[8] Adnane Addaim , Driss Gretee and Abdessalam Ait Madi, "Enhanced Box-Muller method for high quality Gaussian Random Number Generation", Journal Science and Mathematics, Vol. 9, No. 3, 2018

[9] Thomas, DB, Luk . W., Leong, PHW, and Villasenor, JD 2007. "Gaussian random number generators", ACM ComputSurv.39,4, Article 11 (October 2007) Doi:10.1145/1287620.1287622

[10] DP Kroese, T. Taimre, ZI Botev, "Handbook of Monte Carlo Method", Wiley Series in Probability and Statistics, John Wiley and Sons, New York, 2011

[11] Lemire, Daniel, "Fast Random Integer Generation in an Interval", ACM Transactions on Modeling and Computer Simulation, Vol.29 No.1, pp. 1–12, 2019, https://doi:10.1145/3230636

[12] Zulfikar, "FPGA Implementation of Uniform Random Number on Residual Method", Journal Manipulation Elektrika Vol.11 No. 1, 2014

[13] "RM0090 Reference Manual STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs @ https://www.st.com/resource/en/reference_manual/DM00031020. pdf", [Online].Available: https://www.st.com/resource/en/reference_manual/DM00031020-.pdf, February 2021

DOI: https://doi.org/10.29207/resti.v6i2.3684