SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 13, No. 3, October 2016, 361-368

UDC: 004.231.3:621.391.822 **DOI: 10.2298/SJEE1603361F**

Low-Cost portable TRNG, Implementation and Evaluation

Igor Fermevc[1], Saša Adamović[1]

Abstract: This paper will show one of many possible hardware implementations of random sequence generators and give a short survey on existing work related to techniques used for producing true random bits. By using cheap electronic components found in every specialized store such as 8-bit RISC microcontroler, double analogue comparator chip and USB to RS232 interface integrated circuit, we were able to produce a low cost, higly portable device that outputs random sequences with excellent statistical characteristics and high entropy. The source of randomness is a mix of techniques such as electronic noise, phase noise and oscillator jitter. The device in question has a built-in debiasing algorithm similar to [1] and a security mechanism that protects the end user by constantly monitoring the quality of digitized noise signal. Finaly, we will show the results of comparative analysis of data acquired from our device and „random.org“ online service.

Keywords: Randomness, Electronic noise, Free running oscillator, Cryptography.

1 About Randomness

Unpredictable and unrelated occurrences in nature or in a game of cards – randomness, chance or something else? Throughout the history, we have tried to explain, determine, predict or find a pattern in everything that surrounds us. Without going into religious or philosophical discussion on chance and randomness [2], we will stay in the field of mathematics and use Probability theory and Statistics to evaluate the level of randomness in data produced by random sequence generators. The assumption that only physical processes that occur in nature are totally unpredictable, unrelated or random, drives us to study and model the sources of randomness in such a way that they produce results which will be close to ones defined in [3]. By observing the wideband electronic noise signal, we can agree that its characteristics fulfill mathematically defined criteria for randomness, such as normal probability distribution of samples and uncorrelated sampled values. To prove this initial hypothesis in practice, we decided to produce a device that operates on these principals.

[1]Singidunum University, 32 Danijelova Street, Belgrade, Serbia;
E-mails: igor.fermevc.12@singimail.rs, sadamovic@singidunum.ac.rs

2 Previous Work

If we focus on possible types of randomness sources, as given in the survey on hardware random number generators [4] we can divide them in groups as shown in Fig. 1.

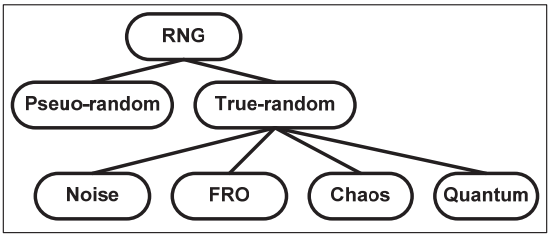

Fig. 1 – *Types of randomness generators*.

Although there is a large number of scientific papers and patents on the subject of hardware randomness generators, there is only a few practical implementations conforming to standards given by the NIST, BIS and other regulatory entities. State of the art generators known as quantum randomness generators are described in [5] and their price is in the range between one thousand and the tens of thousands USD which makes them inaccessible to most academic institutions. Next in line are generators working on principles described in [6] and we can summarize their block diagram of operation in a form shown in Fig. 2.

Fig. 2 – *Block diagram of operation.*

Our device follows the above block diagram and represents a mix of FRO and Noise based group of generators. Concerning its low price and characteristics it stands side by side to commercially available devices shown in Fig. 3. but as opposed to these there are no drawbacks to using unstable PN junction breakdown principle and no need to boost the voltage from USB power rail.

The comparative data for these devices are given in **Table 1**. As already stated, our device operates similar to a free running oscillator with variable pulse length and amplitude. These variations in signal are comparable to wideband white noise signal which gives us a good starting point for further processing.

One common feature with commercially available devices is USB interface which is accepted as a good idea because it allows the device to be portable and locally powered from the host computer. From the security and cryptographic perspective, the USB interface integrated circuit is the only part of the device that we must delegate the trust because we do not have the possibility to fully check all integrated components and features. One way to avoid this uncertainty is to implement USB stack and interface in microcontroller which would be a part of future work.

Table 1

TRNG characteristics.

Model	Speed [Kbs]	Randomness source	Power consumption [mW]	Price [USD]
Alea II	100	PN junction	250	150
TL200	2048	PN junction	100	200
True RNG2	350	PN junction	No data	50
One RNG	350	PN junction/RF noise	No data	50
Our device	72	FRO + El. noise	<100	20

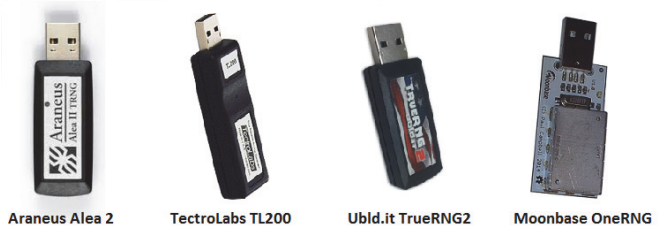

Fig. 3 – *Commercially available RNGs.*

3 Implementation and Evaluation

3.1 Hardware Structure

The starting point for hardware development is the circuit diagram in Fig. 4. We have used the SMD electronic components with the intention of reducing the overall size of the device. The circuit is divided in segments that mimic the previously shown diagram. We start with a double analogue comparator LM393 which acts as noise source and digitizer. Microcontroller Atmel Attiny85 whose running code is written in AVR Assembler and processes the digitized signal, manages communication protocol and maintains the randomness pool. Final hardware segment is FTDI FT232RL integrated circuit used for interfacing the device trough virtual serial port on host computer. We must point out that in order to evaluate certain security mechanism implemented in MCU running code all hardware safety measures have been avoided in the design. In order to use this device in real world

applications, proper EMI shielding and additional power filtering circuits must be installed.

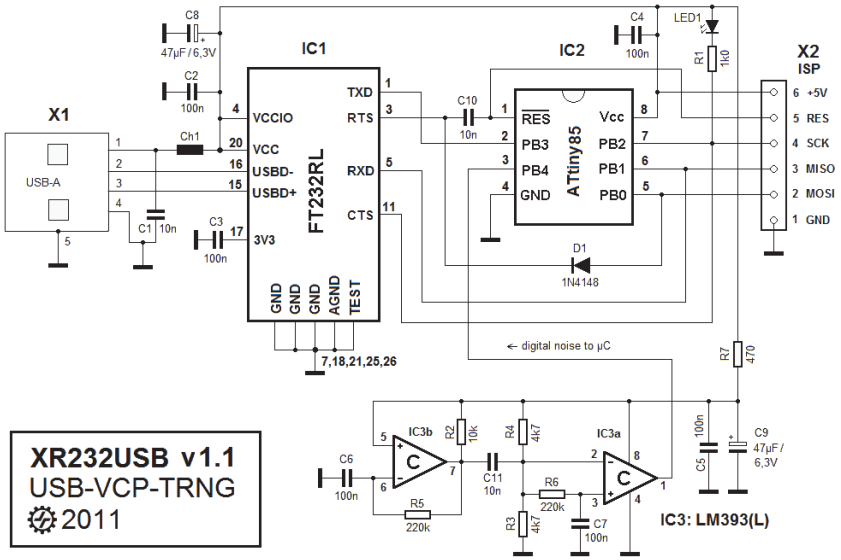

Fig. 4 – *Circuit diagram [8).*

3.2 Device Operation

Double analogue comparator LM393 serves two purposes. The first part of comparator is the source of nonlinearity. As stated in [7], in order to minimize multiple transitions in output when input signal is close to comparator threshold level, special attention must be paid while designing high speed comparator circuits. The right amount of hysteresis, feedback, good grounding, PCB layout, signal routing and decoupling are recommended techniques needed to avoid comparator output ringing. Opposite to best practices, lack of decoupling capacitors, possible stray capacitance and values of electronic components surrounding the first part of comparator chip resulted in, normally unwanted, erratic oscillations of output. Further examination of output signal demonstrated potentially good random characteristics. The snapshot of noise signal acquired by DSO with its main characteristics is presented in Fig. 5.

The second part of comparator works in standard mode. It compares the level of noise signal with reference voltage, amplifies the difference signal and converts it to a digital signal with variable period which is further routed to designated input of MCU. The simplified workflow diagram of MCU operation is shown in Fig.6.

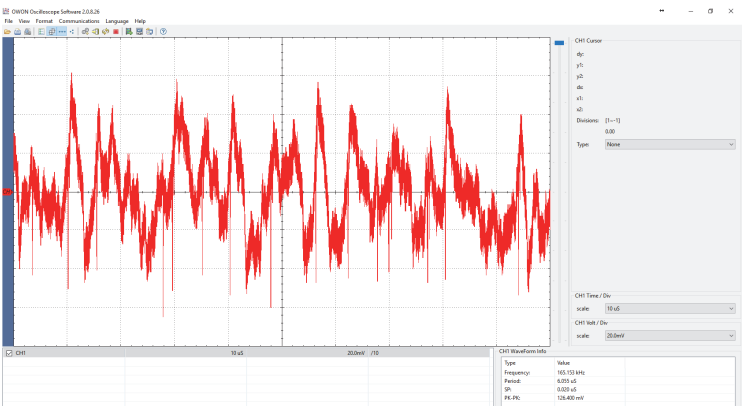

Fig. 5 – *Wideband noise signal from the first part of comparator.*

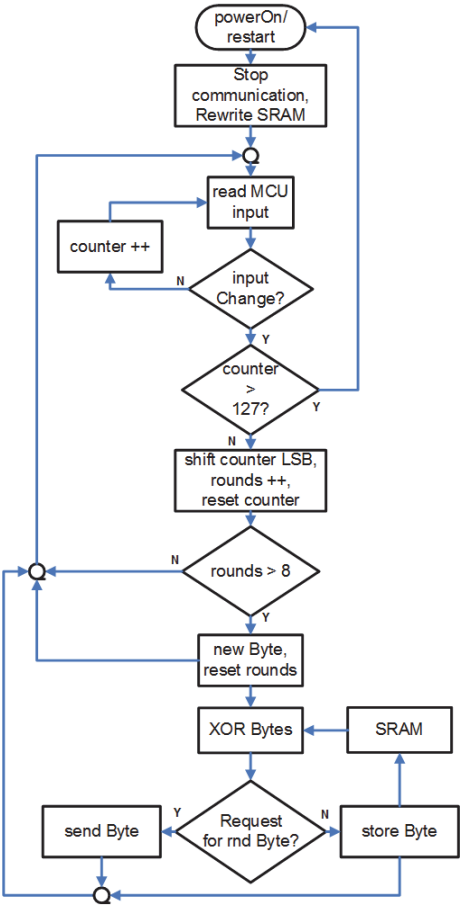

Fig. 6 – *Simplified MCU workflow algorithm.*

MCU running code is organized using macros. There is a separate macro for each state of RS232 handshake signals. Each macro executes the code for randomness pool maintenance incrementing the value of dedicated 8-bit register used as fast binary counter. By monitoring the digitized noise signal value on appropriate MCU input pin, counter is incremented until change in input signal is detected and the value of LSB in counter register is shifted into temporary register. We repeat this process until the whole byte is formed. The newly formed byte value along with previously stored byte is subjected to XOR operation. The resulting value is written into dedicated memory organized in a form of stack. The result of this operation is constantly refreshed pool of 500 random bytes. Because the value of binary counter is dependent on input signal period, if it exceeds certain value, we can assume that generated bytes are not random and utilize this information to activate security macro which stops the serial communication and executes the MCU restart routine followed by emptying and refilling randomness pool. The time needed for MCU to distinguish random from nonrandom signal is not fixed and depends on many factors such as MCU clock speed, noise signal frequency and state of RS232 signals.

3.3 Test Results

Evaluation of device operation is done using software application called XR232 [8] which implements the possibility to pull any number of random bytes from device and store them in a file. Collected data are then submitted to NIST statistical tests described in [9]. Apart from statistical test results, good properties of randomness can also be visualized as lack of pattern as shown in a Fig. 7.

Fig. 7 – *Bitmap of random data sample.*

We have also compared the statistical properties of the sequences produced by our generator and sample collected from “random.org” service, each

consisting of 16 KB. The limitation in the amount of data is imposed by "random.org" service and the results are presented in **Table 2**.

Table 2
Comparative analysis of a couple of NIST test results ($P > 0.01$).

Test type		Our device	Random.org
Runs test (P)		0.439	0.0353
Serial test (P)	(P1)	0.390	0.087
	(P2)	0.858	0.665
Entropy test (bit)	Monobit	0.99999670	0.99997561
	Bigrams	0.99998785	0.99996795
	Trigrams	0.99998438	0.99996382
	Matrix 4x4	0.99997163	0.99995819

The results of statistical tests and comparative analysis proved our initial hypothesis that our device generates truly random bytes. To measure the speed of generation and delivery of the device we conducted an experiment a number of times in which we requested 1 MB of random data while measuring the time. Overall time was 114 seconds so we can conclude that device speed is **72 kbs**. We have also simulated external interference by activating smart phone GSM or Wi-Fi transceiver near our device. If external disturbance is shorter than a couple of milliseconds, as measured with DSO and logic analyzer during the testing, data is delivered from randomness pool without interrupts in communication. Otherwise, the security routine is activated preventing delivery of no random, correlated data to user. Physical appearance of the device is shown in Fig. 8.

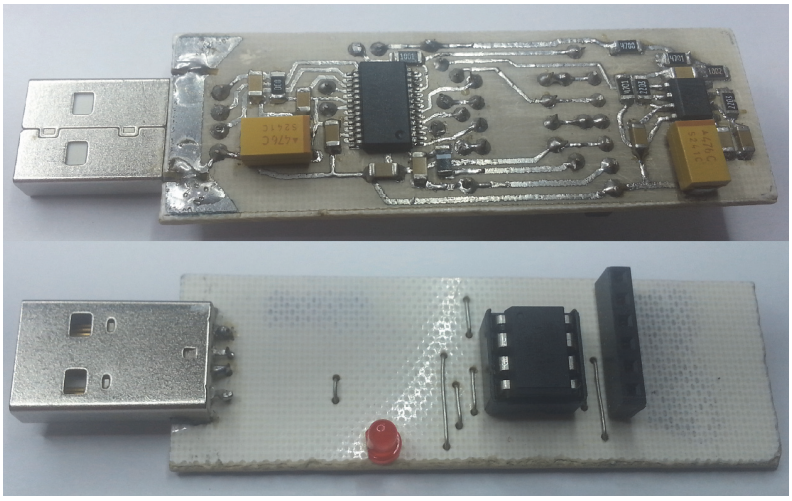

Fig. 8 – *Physical appearance of our device.*

4 Conclusion

We presented a theoretical concept of utilizing electronic noise as a source of randomness and constructed the hardware device to practically evaluate the properties of sequences it generates. We have also verified the quality of randomness source of our device by comparing its results to atmospheric noise based randomness generator. The results of statistical analysis confirmed extremely high level of entropy and unpredictability in produced sequences. We have also implemented and verified a real-time user protection security feature that enhances the trust in our device from cryptographic perspective. Compared to current state of the art quantum random generators the presented device is significantly slower but easy to produce, affordable and usable for various purposes, mainly as an educational tool. With minor modifications regarding EMI shielding and an increase in speed of delivery it can also be used in real life cryptography applications for generating sequences used as session keys for PGP, as strong passwords generator and many more.

5 References

[1] J.V. Neumann: Various Techniques used in Connection with Random Digits, National Bureau of Standards Applied Mathematics Series 12, 1951, pp. 36 – 38.

[2] A. Eagle: Chance versus Randomness, The Stanford Encyclopedia of Philosophy, 2016. Available at: http://plato.stanford.edu/entries/chance-randomness/.

[3] W.B. Davenport Jr., W.L. Root: An Introduction to the Theory of Random Signals and Noise, John Wiley and Sons–IEEE Press, NY, USA, 1987.

[4] M. Stipcevic, C.K. Koc: True Random Number Generators, C.K. Koc. (Ed.), Open Problems in Mathematics and Computational Science, Springer, 2014, pp 275 – 315.

[5] J.F. Dynes, Z.L. Yuan, A.W. Sharpe, A.J. Shields: A High Speed, Postprocessing Free, Quantum Random Number Generator, Applied Physics Letters, Vol. 93, No. 3, July 2008, 031109.

[6] W. Liao, M. Wan, K. Dai, X. Zou: Scalable Truly Random Number Generator, World Congress on Engineering, London, UK, 01-03 July 2015, Vol. 1.

[7] R. Moghami: Curing Comparator Instability with Hysteresis, Analogue Dialogue, Vol. 34, No. 7, 2000, pp. 30 – 32.

[8] J. Thomas: XR232USB - True Random Numbers @ USB.
Available at: http://www.jtxp.org/tech/xr232usb_en.htm.

[9] C. Kenny: Random Number Generators: An Evaluation and Comparison of Random.org and Some Commonly used Generators, The Distributed Computing Group, Trinity College Dublin, Ireland, April 2005.