RANDy - A True-Random Generator Based On Radioactive Decay

Markus Rohe
(mail@markus-rohe.de)

Fortgeschrittenenpraktikum

Advisor: Ammar Alkassar
Security and Cryptography Research Group
Saarland University

SS 2003

Abstract

This paper presents a physical random number generator for mainly cryptographical applications based on alpha decay of Americium 241. A simple and low-cost implementation is shown to detect the decay events of a radioactive source often found in common household smoke detectors. Three different algorithms for the extraction of random bits from the exponentially distributed impulses are discussed. In the concrete application a speed optimized method was chosen to gain a reasonable high data rate from a moderate radiation source (0,1 μCi). To the author's best knowledge this technique has not been applied so far in the context of radiation-based random generators. A tentative application of statistical suits of tests indicates a high quality of the data delivered by the device.

1 Introduction

Today random numbers are employed as well for numerical simulations and computations (Monte-Carlo simulation) as for cryptography. In fact the security of a cryptographic system depends essentially on both the applied algorithms and the quality of the employed random data. During the computation of cryptographic primitives like nounces, challenges or session keys the generation of the random component is considered as a *non-algorithmic* procedure. This character is achieved in practice on several security levels depending on the sort of the applied generator:

1.1 Classification of Random Number Generators

Random number generators can be divided into three classes:

- A *True physical random number generator* (TRNG) is a special piece of hardware that extracts its information by the observation of a stationary physical phenomenon that is provably secure. As quantum theory is intrinsically random[1], a quantum

[1] according to today's awareness

process like thermal noise in a semiconductor or the radioactive decay of an atomic nucleus provides an ideal base for a TRNG. Moreover, it meets the definition of information-theoretic security in cryptography: An attacker is unable to predetermine the bit sequence even with unbounded memory and time resources. Consequently, the use of an unconditionally secure system only remains information-theoretic secure if its indeterministic functions are founded on unpredictable random values.

- A *pseudo random number generator* (PRNG) is a deterministic algorithm implemented either in hardware or in software that generates a long-periodic sequence of numbers that appear to be random according to several statistical tests. The sequence is generated starting from an initial input, called *seed*, usually delivered by a much slower TRNG. Interpreted as a finite automaton the seed indicates the initial state and for every generation step the internal state is changed while the next pseudo random value is computed. Compared to the length of the output the seed is very short so the only protection against an attacker is the lack of computational power or memory that leads from a large enough piece of the sequence to a corresponding state in the algorithm. But if this algorithm is known[2] and one internal state or even the seed is reconstructed, the whole pseudo random sequence can be reversed by an attacker. Hence, a PRNG cannot be regarded as unconditional secure any more. With well-chosen parameters and algorithms they meet the requirement of *computational security*.
- The third class of random number generators are PRNGs whose seed is not generated or regularly updated by a true-random source. Instead the seed consists of data derived by chaotic and complex phenomenons like the current head position of the hard disc or other OS-parameters. This data appears to be random because a discrete computation or reconstruction is still impossible nowadays. Thus, this type of RNG even uses a seed that is only computationally secure. In some cases there have been some poor choices of seed [8], admittedly: The SSL-session keys in former Netscape browsers were easy to calculate for an attacker. Devices that observe directly such a chaotic process like disturbances in radio broadcasting to gain "random data" fit the same category.

Consequently: To obtain provable random data there is no alternative to a TRNG.

1.2 Related Work

Many TRNG devices basing on quantum processes have been constructed so far, both for research and commercial purposes. Here a small choice is listed among many others: *Clipped white noise* gained by thermal noise of resistors or semiconductors is the base for the designs by [23] and [20]. [29] extracts it even from neon tubes. This noise is amplified, evaluated by a comparator, sampled and digitally processed to suppress correlations and statistic errors. An alternative approach is to modulate a voltage controlled oscillator by a noise voltage. Its output is compared to a stabilized oscillator (VCO) [16] that runs much faster. Each time a transition occurs in the VCO the current value of the fast running oscillator is interpreted as the next random bit. There are also physical random number generators available based on optical quantum processes [28] that allow fairly high data rates. Finally radioactive decay [12] [6] can also be evaluated. The advantage of implementing the latter phenomenon is the fact that it provides fairly good detectable single events either with a Geiger-Müller tube or a semiconductor sensor. This is less error prune than amplifying and processing the noise of e.g. a semiconductor element which is very sensitive to high frequent electromagnetic disturbances [23]. On the other

[2] if the undesirable policy "security by obscurity" is not practised

hand one always has to deal with a radioactive substance so it is a little more difficult to handle than a simple circuit with a special noise diode.

1.3 Overview

I section 2 a practical method is chosen to implement a quantum process. Afterwards the theoretical background of the radioactive decay is stated and 3 algorithms to extract random data from this phenomenon. Section 4 presents the entire TRNG device with implementation details. A preliminary statistical evaluation emerges a high quality of some collected data and finally listings, layouts and some photos are enclosed.

2 Preconsiderations and Abandoned Ideas

At the beginning of this project several aims and demands to the device were set:

- High quality of random data for cryptographic application
- Random data ready for use, no software transformations necessary
- Fast enough for a single user application
- RS 232 interface for maximal independence to the operating system
- Solid design but not too expensive
- Easy to reproduce, standard components and no special developing devices or tools required
- A certain immunity against environmental influences and attacks.

Besides of these requirements three methods were suggested to extract the random bits at the beginning of this project.

- Radioactive decay: *Using radioactive material from a common household smoke detector without a sensor*[3]. The decay impulses should be detected by the conductivity of the air between the electrodes. Each time an alpha particle rips through the air it stripes off electrons of nearby atoms with enough force to cause these electrons to ionize other atoms. Applying a voltage[4] on both electrodes the set of ions travels through the electric field and discharges at the metal plates generating a small current pulse. This would have been a very elegant and inexpensive solution but the amount of ionized particles was too small to generate a reasonable high current pulse even on a high-quality digital oscilloscope. These very low currents are difficult to handle: There are amateur radioactive measurement devices based on the ion chamber principle [35] but the circuits are quite instable and very susceptible to static electricity. Wenzel uses a peanut can, very high impedance FETs and feedback resistors of about 100000 MΩ to amplify the currents in the region of about 0,01 fA.
  Measurements by Rosing [24] show that no exact pulses are visible. In his random generator he employs the detection chamber as a source of nearly white noise. A fourier analysis of the amplified signal is almost flat. This is not quite surprising

[3] There are mainly two types of smoke detectors available: optical and ionization type. The latter one uses a plastic capsule with two horizontal metal plates. At the bottom plate a small amount of americium 241 with an activity of 1 μCi (33 kBq i.e. 33000 decays per second) is fixed. The second plate is mounted about 5mm above the source with a hole in it to allow the air to filter in. The air between the electrodes is weakly ionized by the alpha particles so a small current flows between the electrodes. If smoke particles enter the chamber they absorb many charged air molecules. As a consequence the conductivity of the atmosphere sinks and the alarm is triggered. An interesting article with some photos is given here [3].

[4] no high voltage necessary

because at atmosphere pressure the ionized air molecules move at speed of sound to the electrodes for discharging; the release of the signal becomes too long. And since many ions are diffusing through the chamber air, no sharp pulses should be visible.

- The second option: evaluating the *clipped white noise* of a semiconductor device as a base for random bits. This was already implemented by [1] for low cost properties and is a quite common method [23] [12] [5]. Thus, there would not have been many new aspects to discover.
- *Radioactive Decay:* Use a radioactive source with a semiconductor sensor or a Geiger-Müller tube to obtain stable impulses. The length of the time intervals between two consecutive decay impulses is unpredictable. This method was applied to construct this TRNG. RANDy bases on a preparation of the alpha radiator Americium 241[5] from a common household smoke detector so the total amount of radioactive material is not very critical[6]. Geiger Müller tubes are rather common for simple qualitative measurements of mostly β-radiation. On the other hand, a semiconductor sensor has the advantage that no high voltage is necessary and the recovery time after an inpulse was detected is much smaller than in a tube. For quantitative radiation measurement and radiation monitoring PIN-diodes are widely used[7]. [13] even constructed a wideband radiation monitor with a standard PIN photo diode where the glass window was removed. This circuit idea has become a part of the pulse detector. The sensor is a standard optical PIN diode BPX61 [27] without glass window. It is mounted directly above the radioactive material of the ion chamber and generates very accurate pulses. The blank diode is sensitive to any radiation above a certain wave length (infra red, visible light, X-ray, α, β or γ-quants). But since both the diode and the radioactive material are placed inside a metal box only the alpha particles can cause an ionization event (see photos in Appendix C).

3 Theoretical Background of the Bit Extraction

The TRNG presented in this work consists of a radioactive source and a corresponding detector. The decay impulses are filtered and amplified for a further digital processing. The random bits are obtained by deciding whether the time interval between two pulses consists of an even or odd amount of timing units. This processing is done by a microcontroller that sends the random data via RS232 to a host computer where it is captured by a standard terminal program and stored or used.
The following section gives a more detailed description of the statistic modelling of the radioactive decay. Furthermore, three methods to extract random bits from short pulses generated by radioactive decay in are discussed. These algorithms are also suitable for pulses generated by zero crossing events of a noise voltage as e.g. in [23] [12]. The principal aim is to obtain a correlation-free bit stream that holds the equal distribution, i.e.

[5] Energies: α: 5 MeV, γ: 25 keV and 60 keV, half life about 433a [2]

[6] In fact the alpha particles emitted will not even penetrate a sheet of paper. Although one has to keep in mind that there is a source of radiation after all that can be spread into the environment due to a fire or a mechanical accident. The toxic effect of americium is much more critical on human organism when incubated caused by its heavy metal quality in general. It is accumulated especially in the bone material where the low range but high ionizing *alpha*-radiation is damaging the organism directly from inside [2].

[7] In a p-i-n diode the pn-junction (deplated region) is extended by a thin layer of undoped (intrinsic) material between the p and n type regions. In this area hole-electron pairs are formed when light or other forms of ionizing radiation is absorbed. Thus charge Q can be accumulated: $Q = \Delta E/\varepsilon$ where ε is the ionizing constant of silicon and ΔE is the energy that the ionizing event accumulates in the active intrinsic region of the diode [13]. If the energy of the ionizing quant can be completely absorbed by the silicon it is proportional to the generated charge Q. Under this circumstances quantitative measurements become possible. For more detailed information on PIN diodes and photodiode amplifiers: [10] [4]

$P(0) = P(1) = 0,5$[8].

3.1 Some Properties of the Radioactive Decay

The behaviour of the radioactive decay is mathematically modelled by the *Poisson distribution*, for further information see [14]. The corresponding *distribution of distances* between consecutive decay events is the negative *exponential distribution*.

$$p(t) \;=\; \lambda e^{-\lambda t} \;\; (t \geq 0)$$

where λ is the intensity parameter of the process and the mean value $\mu = \frac{1}{\lambda}$. $p(t)$ describes the random time distance between two successive impulses. Two preconditions must hold:

- The amount of decaying atomic nuclei is large enough to be considered as constant during the measurement time.
- The half life of the isotope is large enough and therefore the intensity parameter λ remains constant.

Both conditions are satisfied in the concrete application. The probability (cumulative distribution) that a decay event X occurs within the time interval $[0, t]$ is

$$P(0 \leq X \leq t) \;=\; \int_0^t \lambda \cdot e^{-\lambda x} dx \;=\; 1 - e^{-\lambda t}$$

while the probability that during the interval $[0, t]$ no impulse occurs is

$$P(X > t) \;=\; 1 - P(0 \leq X \leq t) \;=\; 1 - (1 - e^{-\lambda t}) \;=\; e^{-\lambda t}$$

Another important aspect of the exponential distribution is the *memoryless property* or *Markov property*. It states that the distribution of the distance between two consecutive event points is the same as the distribution of the distance between an arbitrary chosen point and the next event point. This can easily be justified: Fix an arbitrary time point T and a time interval Δt. Then the probability that the next event point occurs after $T + \Delta t$ is independent of the choice of T

$$\begin{aligned} P(X \;>\; T + \Delta t \mid x > T) \;\; &= \;\; \frac{P(X \;>\; T + \Delta t) \;\cap\; P(X \;>\; T)}{P(X \;>\; T)} \\ &= \;\; \frac{P(X \;>\; T + \Delta t)}{P(X > T)} \quad \textit{since } \Delta t > 0 \\ &= \;\; \frac{e^{-\lambda(T+\Delta t)}}{e^{-\lambda T}} \;=\; e^{-\lambda \Delta t} \\ &= \;\; P(X > \Delta t) \end{aligned}$$

Conclusion: Only the *time difference* matters, it is unimportant *where* this difference occurs. If a half life of e.g. an electronic device or an atomic nucleus is described this way the exponential distribution does not know anything about aging! This is very important because of the dead time of the sensor after having registered a decay it is unsensitive for a short time to register another event. After the end of the dead time there is the same probability to detect the next decay in a given time interval.
Not every particle reaches the sensor and generates an amount of charge because of the distance between the sensor and the substrate and the spherical radiation characteristic.

[8] P(0) is the probability that the following bit extracted from the bitstream is a 0

Also the particles differ in their intensity when hitting the sensor which results in different amounts of charge respectively different peak heights before the comparator[9]. The radioactive desintegration of Americium generates new daughter nuclides that are also radioactive and cause peaks of different height at the input of the comparator. But this does not change the statistical behaviour of the whole process; only the activity factor λ changes over the time according to the *additivity property*:

$$\begin{aligned} \lambda &= \sum_{i=1}^{m} \lambda_i \\ P(X > t) &= e^{-(\sum_{i=1}^{m} \lambda_i)t} \end{aligned}$$

In this sum the intensity parameters λ_i of all statistical independent Poisson processes are united that generate a sufficient high impulse at the input of the comparator.

3.2 Method 1

The generators by Gude [12] and Vincent [30] base on the fact that in a Poisson process the amount of impulses or elementary events within a fixed time interval cannot be predetermined. The probability that k impulses are registered within the interval Δt is

$$P(k) = \frac{(\lambda \Delta t)^k}{k!} \cdot e^{-\lambda \Delta t}$$

Both trigger a toggle flip flop with the decay pulses and evaluate its state after a constant amount of time. Afterwards the the flip flop is set back at the end of the time window to guarantee a new memoryless measurement in the following interval.

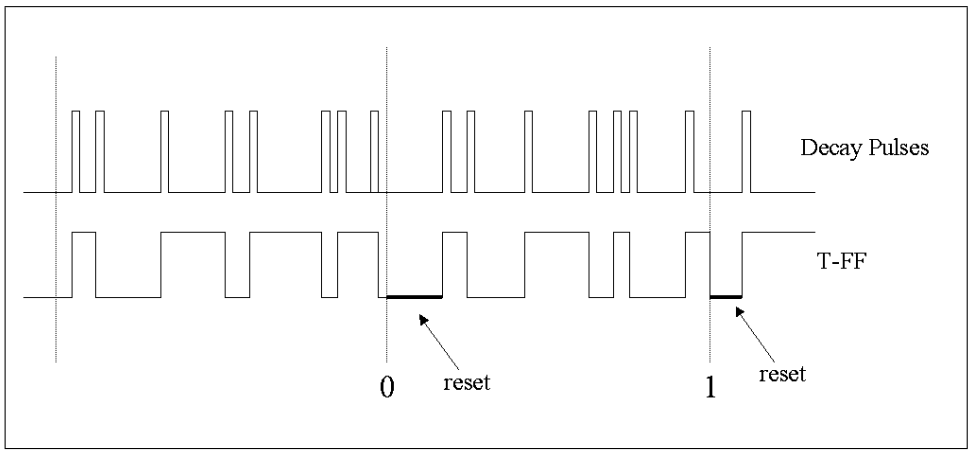


Fig. 1: Counting the pulses within a constant time window

According to Gude [11] the time window should be chosen at least 10 times larger than the mean decay rate μ of the radioactive substance to minimize the influence of autocorrelation and to hold the equal distribution $P(0) = P(1) = 0,5$. These errors decrease exponentially by the additive factor $e^{-2\lambda \cdot \Delta t}$ with increasing time interval Δt[10]. Vincent [30] [31] and Kraus [17] receive the same results. PURAN1 [12] produced considerable true random values implementing this algorithm but a lot of decay impulses are wasted.

[9] The applied Americium substrate is expected to emit about 33000 decays per second. Due to the spherical diffusing characteristic half of then vanish into the ground plate where the substrate is fixed. The common PIN photo diode has a sensitive surface of about $10mm^2$ so we are only able to register about 1700 impulses per second, totaly about 5%.

[10] $P(0) - P(1) = e^{-2\lambda \cdot \Delta t}$

If one uses a low level radiation source instead of a more active laboratory substance the gained rate of random bits would be too low.

3.3 Method 2

Another approach is used by Walker's "hotbits" [34] and suggested by [6]. They measure the time between two decay events and decide the bits as follows:

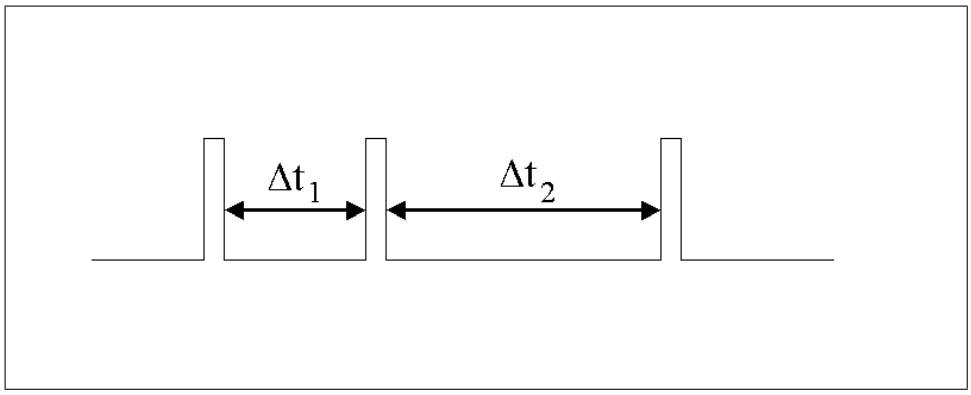


Fig. 2: Comparing the length of two consecutive decay intervals

- $\Delta t_1 > \Delta t_2$: interpret this result as a **1**
- $\Delta t_1 < \Delta t_2$: interpret this result as a **0**
- $\Delta t_1 = \Delta t_2$: discard this event, continue with the next pair of intervals

The exploit of this method is much better than the first one: Two decay events are necessary to extract one bit. But here one needs two timer registers of reasonable size depending on the clock speed and the mean decay rate μ. This might be a problem if a long-term shift in half life occurs and the timers produce an overflow as a consequence. So far, there exist no statistical evaluations of an implementation of this extraction algorithm.

3.4 Method 3

Due to the *memoryless property* the time interval between two consecutive decay events cannot be anticipated. A time measurement between two events is taken with a high resolution clock and classified the following way:

- the length t of the interval consists of an odd amount of timing units Δt:
  $t \equiv \mathbf{1}\ units\ mod\ 2$
- the length t of the interval consists of an even amount of timing units Δt:
  $t \equiv \mathbf{0}\ units\ mod\ 2$

This is the highest gain of information so far: *one decay delivers one random bit.* A small shift in the mean decay rate maybe caused by the aging of the radiation source does not affect the classification and makes this algorithm quite robust. It can be implemented by a toggle flip flop (T-FF) in practice. Each time a pulse occurs, the current value of the T-FF is captured as the new random bit. The T-FF has to be set back to a defined value (e.g. Low). This is analog to a new time measurement where the clock also has to be set back to 0. Hence, each measurement is completely memoryless and therefore uncorrelated [11]. It should be clear that the output-ratio of the T-FF must be exact at 50:50 otherwise the bits would be coloured, i.e. the balance between 0's and 1's would not be equal any more. If this would be the case, however, Appendix B cites a simple method by von Neumann to eliminates this bias in the equal distribution but at the prize

of a much lower data rate; about 75% of the bits would be lost.
A formal discussion of the transformation from the exponential distribution to the binary equal distribution can be found in Appendix A. If the timing units are reasonable small the errors in the equal distribution are negligible, in fact it decreases exponentially by factor $e^{-\lambda \cdot \Delta t}$ where Δt is the length of the timing unit.

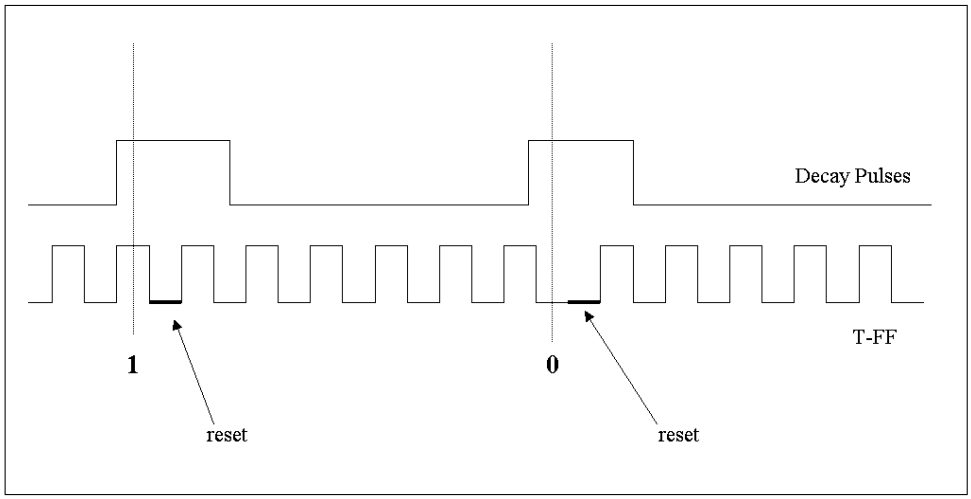


Fig. 3: Time measurement mod 2 between two consecutive impulses.

An error occurs in time measurements if the event occurs shortly after the evaluation point. Then it is registered to the next sampling point and interpreted some time t_{err} later.

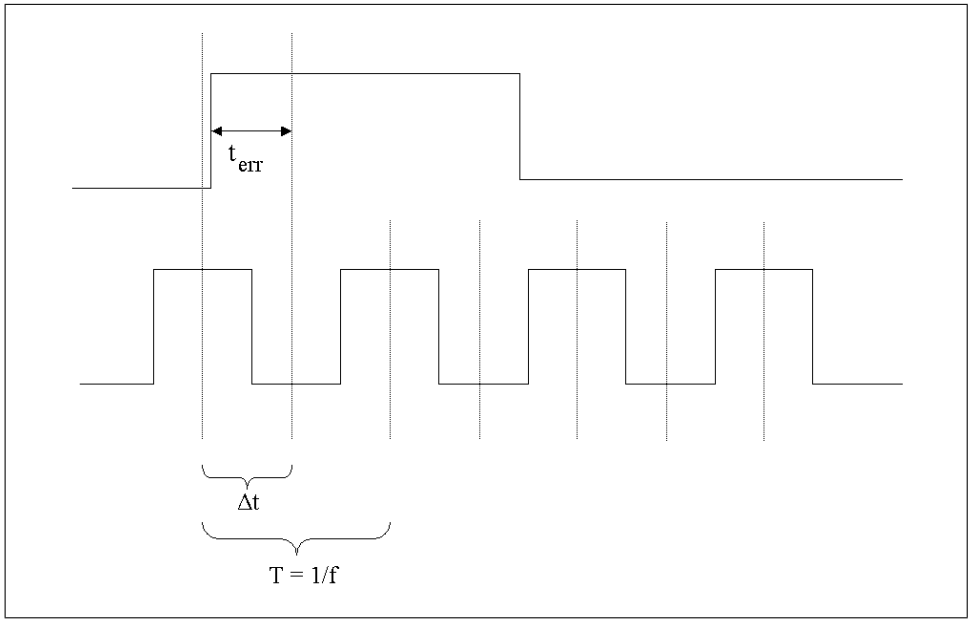


Fig. 4: Error in time measurement

In the worst case, the maximal error is up to one timing unit Δt and affects the measured result linear to the applied frequency f:

$$t_{err} \leq \frac{T}{2} = \frac{1}{2 \cdot f}$$

If one suppose a equal distribution of this error length due to the unpredictability of the decay intervals this error should not affect the equal distribution for the sake of the large clock ratio[11]. To the author's best knowledge this method has not yet been implemented in any radiation based TRNG and some preliminary statistical evaluations indicate a high quality of the obtained random data. Even symmetration (see Appendix B) appears to be unnecessary.

3.5 Further Problems

Other imperfections of the electronic components e.g. propagation delays of the flip flops and clock jitters are hard to estimate and can only be minimized by using high speed components. But these effects occur in each implementation of a sampling and will never be eliminated completely. The "ideal" hardware random number generator will never be possible. But with better and better components it will become hard to collect enough random data to discover these weaknesses.

4 Description of the Circuit

The complete TRNG mainly consists of an analog and a digital part. In the analog section the decay pulses are detected and amplified. To avoid noise and other perturbations this part is shielded completely in a metal box with battery power to prevent spikes from the mains supply. A low power circuit indicates if the voltage of the battery or accumulator has reached a lower level so a change or rechargement is necessary. Spikes from the digital ICs via supply or ground line are excluded by optical transmission of both the impulses and the low battery signal. In the digital section the decay pulses are proceeded, the random information is extracted and transferred via RS232 interface to a host computer.

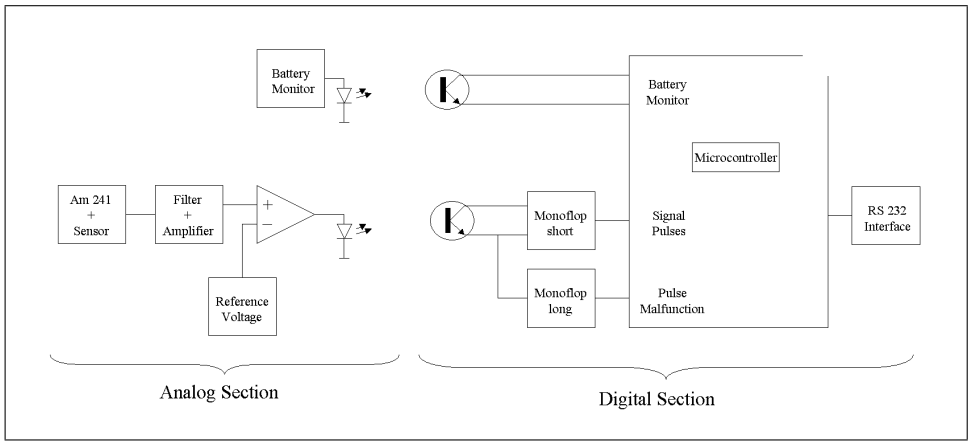


Fig. 5: Block diagram if the circuit

4.1 Analog Part

The reverse voltage of the PIN diode is normally kept as high as possible (up to 40 V) to decrease the capacity and to ensure a fast switching characteristic [4]. In this application the diode is powered by a lower voltage (5V) since experiments have shown that at these unconventional conditions the desired pulses are better visible under the thermal noise floor. The AC-coupling to the integrator IC1A (TLC274)[12] [15] blocks long term DC potential changes and only short pulses can pass. A filter with two capacitors

[11] Certainly this assumption has to be justified by extensive statistical analysis.

[12] The OP-amps were chosen for working at a low single voltage and a very high input impedance

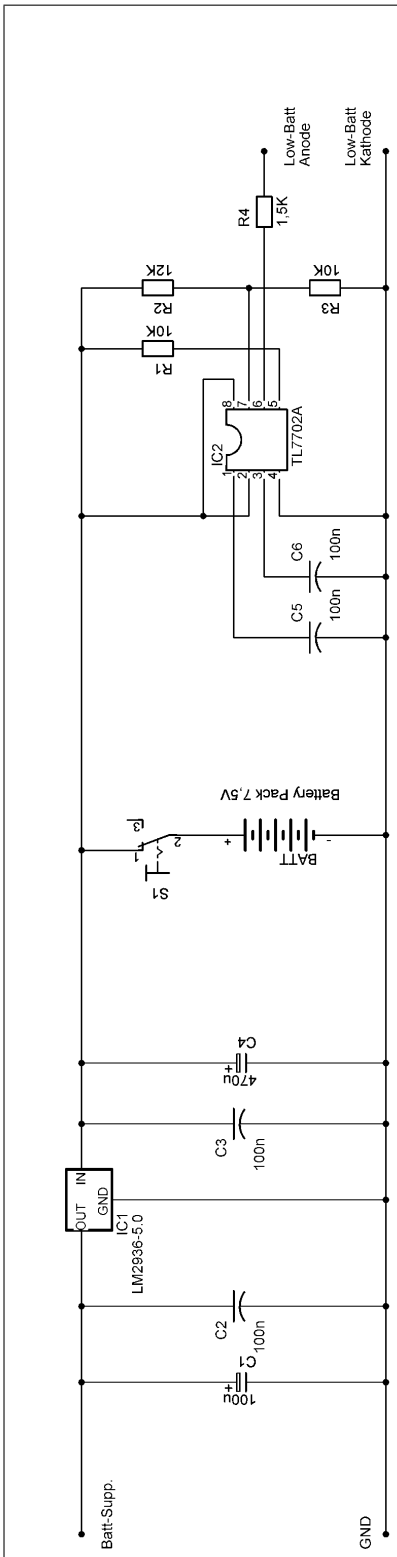


Fig. 6: The Battery Supply

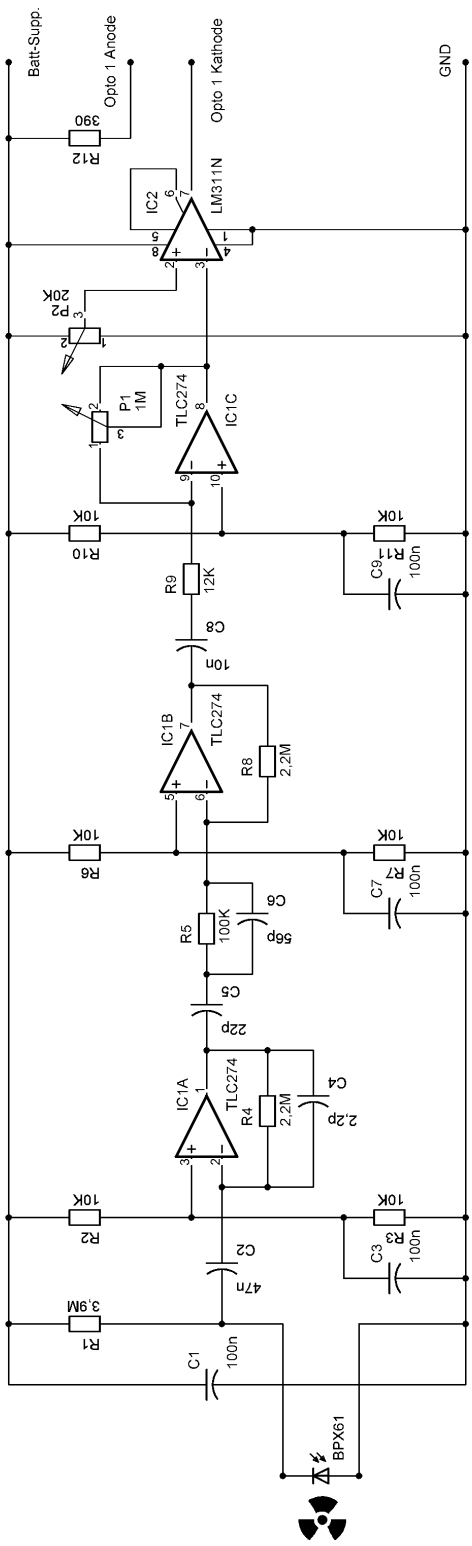


Fig. 7: The Pulse Detector

and one resistor extracts the pulse events from the noise floor before this signal is raised to about 60 mV by the preamplifier IC1B. The third OP IC1C prepares the signal for the voltage comparator by amplifying it again by factor up to 10 depending on the selected value of P1. Finally the voltage comparator LM311 [26] triggers the optocoupler's LED whenever the signal falls below the reference level determined by P2, i.e. a pulse has occurred (see oscilloscope screenshots in Appendix C). The complete unit is quite sensitive, consequently, all voltage dividers and supply lines to the ICs are blocked with ceramic capacitors. The PCB shown in the appendix C is not 100 percent compatible to the working prototype. Actually the 4th OP amp in the TLC274 package was intended to act as comparator. But for every trigger event a second impulse occurred at the input of the entire amplifier construction. The result was that each peak on the output was followed by another peak within a constant time interval. Clearly such a signal quality is unacceptable. For this reason a daughter board with a separate comparator chip LM311 was soldered above the main board. In a redesign step the first two amplifier stages could be constructed with a dual Op-amp TLC272 the main amplifier separated in a single chip (e.g. TLC271) followed by the comparator LM311. An individual shielding of each OP amp group could become necessary.
A low drop voltage regulator LM2936-5.0 [26] with very low own current consumption (several μA) generates a fixed supply voltage of 5V. The TLC7702 [15] monitors the battery voltage and triggers the optocoupler's LED if the voltage falls below 5,5V. This is the lower bound when the supply regulator does not work stable any more.

4.2 Digital Part

The optocoupler 6N137 transmits the pulses from the analog part into the digital section is suited for high speed data transmission (gigabit range, [25]). Hence, the pulse length of about 2-4 μs is within the specification. For the low battery signalisation a Darlington-type opto coupler was chosen (6N139 [25]) so only about 1mA is required from the discharged battery to indicate the need for exchange.
The pulse source is driving two retriggerable monostable flip flops (mono-flop, 74HCT123 [15]). The first one equals the pulse duration to about 7μs to provide a constant signal quality for the evaluation in the microcontroller PIC16F628 [19]. The second mono-flop has a much larger timing constant than the mean pulse rate. During normal operation conditions this mono-flop is permanently set back but if the pulse source breaks down the output changes from high to low and the microcontroller immediately stops transmission. Finally a MAX232 [22] converts the TTL voltage between 0V and 5V to a true RS 232 level ($\pm$12V).

4.3 Evaluation of the Pulses

The pulse extraction and the communication via RS232 to a host computer is provided by the microcontroller PIC16F628 [19]. The free assembler for WINDOWS-platform can also be obtained by the manufarctor. The controller is programmed e.g. with software by Nigel Goodwin [9] and several schematics for programming hardware can also be found there.
The microcontroller is running at 18.432 MHz, a well suited speed to generate internally the clocks for the serial transmission via the internal UART. Instead of a T-FF, the least significant bit of an 8-bit counter register (TMR0) is used. The clock for this register is divided by 4 so it runs at about 4,6 MHz, the base for the time measurements between the pulses on the input pin RA3. Furthermore, the reading of the counter is internally synchronized [19]. This is a counter measure to the jitter effect, mentioned in Section 3.5. The random bits are obtained by method 3. This algorithm adopted to the microcontroller is the following:

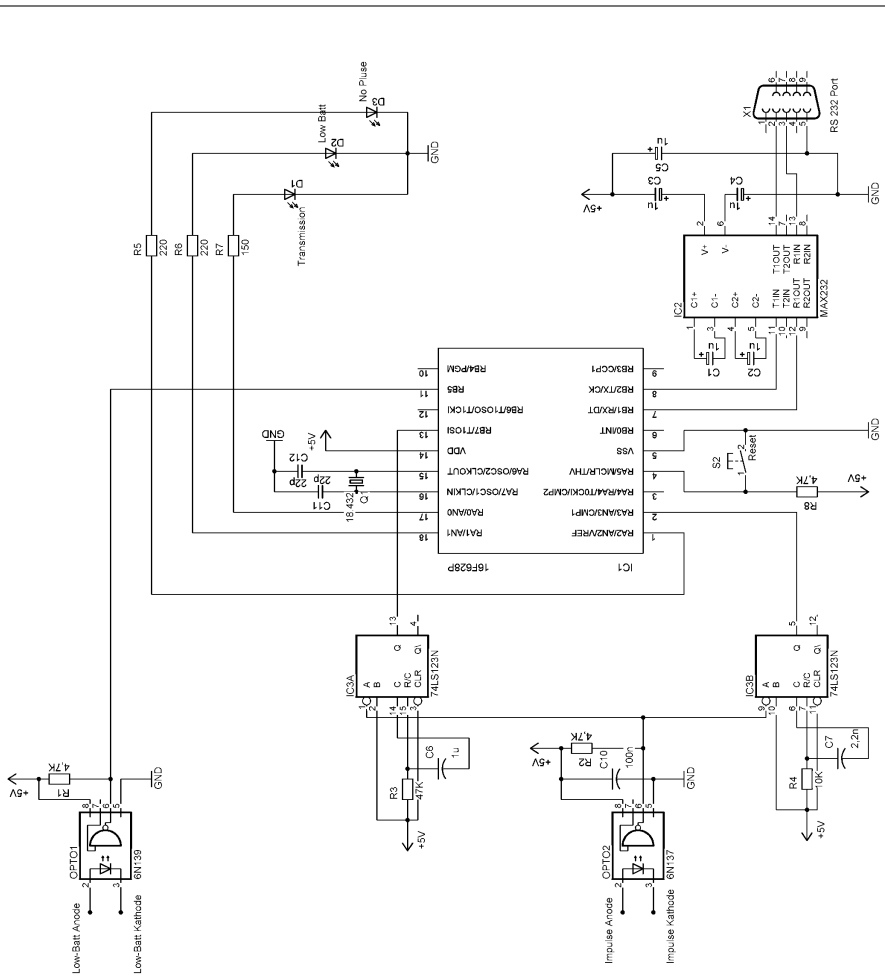


Fig. 8: The Digital Part

- **Step 1:** Poll for high on input RA3
- **Step 2:** Read the counter (TMR0) value, reset it and isolate the least significant bit (lsb) that will be shifted into the random byte
- **Step 3:** Poll RA3 for low and goto **Step 1** if 8 bits are not yet collected
- **Step 4:** If the byte is completed with 8 new random bits it is sent via the internal UART
- **Step 5:** After the transmission the whole capture process has to be resynchronized because during the possible wait time for the UART to be finished several pulses may have been missed
  Resynchronize: poll for low, poll for high, reset counter, poll for low, goto **Step 1**

There are two conditions for the transmission of random data to be instantly aborted because the random quality cannot be guaranteed any more:

- **LOW_BATT:** The supply voltage in the analog section falls below 5,5V. Then the voltage difference is too small for the regulator to form stable 5V. But the reference input of the comparator depends on this voltage via P2.
- **NO_PULSES:** Although the power supply of the analog part seems to work properly no pulses can be detected. This may indicate several serious problems and for the search of the defect an oscilloscope is necessary:
  - mechanical defect of the diode
  - defect of OP-amp, comparator or impulse opto coupler
  - wrong adjustment of P2

At the start of the program the fail conditions are checked and eventually a transmission is inhibited. Later during the evaluation an interrupt is triggered if one of the fail conditions occurs; the transmission is stopped. Practically the feature "interrupt on change of RB4..RB7" of the controller is used [19]. Each time the level of the I/O lines RB4..RB7 changes from high to low or low to high an input an interrupt is triggered. The reason for the interrupt is read out in the interrupt service routine (ISR), so it can be determined whether no pulses are detected (i.e. RB7 went from high to low) or low batt is indicated (RB5 went from high to low). After registering a fail condition the controller stalls in an endless loop where it only can be restarted by pushing the reset button.
The diode registers about 1700 decays per second. About 1600 bits are extracted and transmitted since the controller occasionally has to wait for the UART to transmit the next byte. This leads to an effective data rate of about 200 Bytes per second.

4.4 Some Further Modifications

If a larger amount of random data at certain times is needed, an additional memory could be integrated into the generator.
More and more motherboards in modern computers do not offer a serial interface any more. A support of USB should be considered although additional drivers will have to be provided. But also when using the serial port customized programs instead of capturing the port with a standard terminal client will simplify the everyday usage of RANDy.
The BPX61 has a rather small surface of about $10mm^2$ and only a pulse is triggered if the silicon surface is hit. In fact, only about 1700 pulses per second are registered, a rather low exploit of about 5% is achieved. Hence, the data rate could be increased by using a photodiode with a larger surface.
If a high energy particle hits a semiconductor material it causes a certain damage in the crystalline structure such that the semiconductor qualities become lost. An experiment was made [7] with PIN diodes used in a particle accelerator for heavy ions. The diodes were bombed with protons and other positive ions of much higher kinetic energies than the

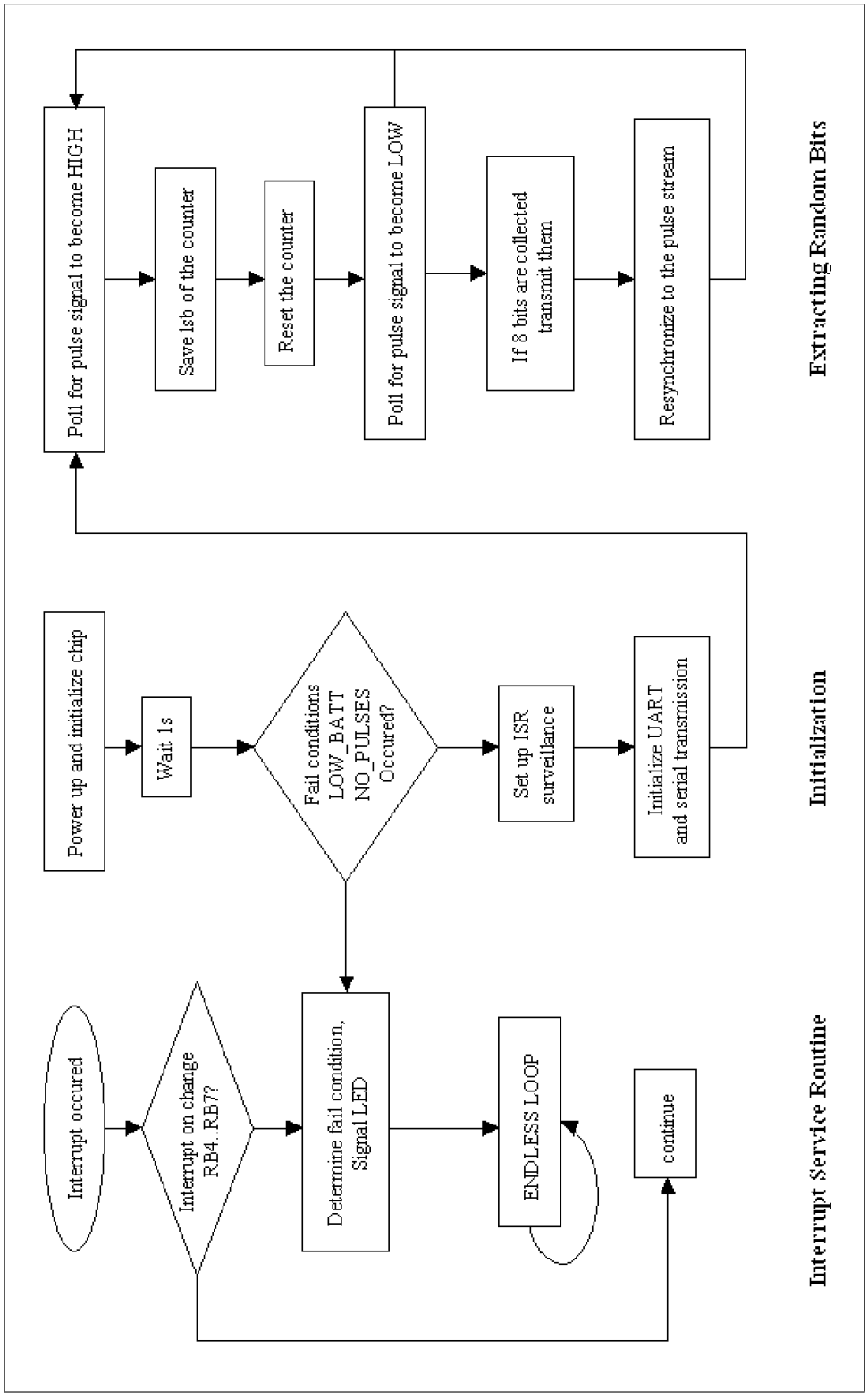


Fig. 9: Diagram of the microcontroller program

α-particles. If one considers only the amount of hitting ions, regardless of their energy and mass, it would take about 20 years to cause the same damage[13] to the embedded diode in RANDy. During its test period of several weeks no changes in the signal quality could be observed but some long time observations have to be made.
Another problem could be the oxidative damage of the silicon surface by air and humidity although the diode material is already passivated [4]. The construction of an encapsulated measure chamber could become necessary. Then its inner space would be filled with a chemical inert gas, e.g. nitrogen or argon.

5 Statistical Evaluation

A preliminary testing was performed on three files consisting of about 11.5 MByte (95,500,000 Bits) each with two free available suits of tests: ent [33] and Diehard [18]. A third battery by the NIST [21] has not yet been applied. Still an extensive examination of the generated data is necessary but the provisional results indicate a high quality.

5.1 Uniform Distribution

A simple counting indicates whether the amount of zeros and ones in the bitstream is uniformly distributed over the data set. According to these results an unbiasing by von Neumann (see Appendix B) appears unnecessary.

File	Bits total	0	%	1	%
1	99,051,928	49,526,448	50.0005	49,525,480	49.9995
2	92,157,560	46,077,941	49.9991	46,079,619	50.0009
3	91,910,032	45,957,576	50.0028	45,952,456	49.9972

Table 5.1

A concatenation of all three data sets leads to a distribution rate of 50.00078%. PURAN2 [23] achieves 50.00092%.

5.2 ent-Suite of Tests

This package contains 5 tests to judge the quality of random data: The mean value (uniformly distribution), the entropy (per bit), the χ^2 - Test,a Monte-Carlo simulation for π and the serial correlation coefficient. For further information about these tests see [33]. RANDy's data achieves maximal entropy per bit (1.000000), an unsuspicious χ^2 - value between 25% and 75% and excellent values for Monte-Carlo π and the serial correlation coefficient.

File	Entropy	p-value χ^2	simulated π	Error %	serial correlat. (opt: 0.0)
1	1.000000	75	3.142597262	0.03	-0.000030
2	1.000000	75	3.141881373	0,01	-0.000048
3	1.000000	50	3.140451809	0,04	-0.000006

Table 5.2

[13] increase of leakage current and decrease of photocurrent

5.3 Diehard Suite of Tests

Diehard [18] is a powerful and extensive battery of tests including 234 single tests. It is performed over a data file of at least about 11.5 MBytes and outputs 234 p-values, one for each test. These values have to be equally distributed over 0 and 1. The supplementary description states that a test "fails big" if one receives values very close to 0 and 1 like 0.0012 or 0.9983. In fact, 6 or more values of 0 and 1 are necessary to fail this test[14].

p-value range	occurences in file 1	occ. in file 2	occ. in file 3
0.0 - 0.1	24	20	25
0.1 - 0.2	25	22	24
0.2 - 0.3	24	19	22
0.3 - 0.4	18	29	37
0.4 - 0.5	28	20	16
0.5 - 0.6	17	25	24
0.6 - 0.7	23	19	28
0.7 - 0.8	27	28	17
0.8 - 0.9	21	28	15
0.9 - 1.0	27	24	26
minimum p	0.0041	0.0001	0.0060
maximum p	0.9996	0.9938	0.9991

Table 5.3

Table 5.3 contains the smallest and the largest p-value of each data file. According to this interpretation, the random data passes the Diehard tests.

6 Conclusion and Outlook

In this paper a TRNG based on radioactive decay was presented for mainly cryptographic applications. The extraction of random data was achieved by applying a fast evaluation algorithm - *one decay impulse leads to one random bit* - so with a moderate radioactive source even a home usable device was created that delivers a reasonable high data rate. It should be possible to gain more than one random bit per time interval between two decay events if one uses a modulo N counter instead if a toggle flip flop (formally a modulo 2 counter), such that a further increasing of the data rate is possible. According to [29] the error in the equality of the distribution should behave linear to the applied sampling frequency. Therefore, a higher sample frequency is necessary which could call for high speed components like FPGAs or GaAs.

7 Acknowledgments

I would like to thank the whole staff of the Lehrstuhl für Hoch- und Höchstfrequenztechnologie at the Saarland University, especially Dr. Thomas Nicolay for extensive support

[14]The original preface of each evaluation protocol states (quote):
NOTE: Most of the tests in DIEHARD return a p-value, which should be uniform on [0,1) if the input file contains truly independent random bits. Those p-values are obtained by p=F(X), where F is the assumed distribution of the sample random variable X—often normal. But that assumed F is just an asymptotic approximation, for which the fit will be worst in the tails. Thus you should not be surprised with occasional p-values near 0 or 1, such as .0012 or .9983. When a bit stream really FAILS BIG, you will get p's of 0 or 1 to six or more places. By all means, do not, as a Statistician might, think that a $p < .025$ or $p > .975$ means that the RNG has "failed the test at the .05 level". Such p's happen among the hundreds that DIEHARD produces, even with good RNG's. So keep in mind that " p happens".

and advice in practical implementation of the analogue section of the circuit. Further Dr. Robert Haberkorn for providing information on radioactive materials and isotopes and some radiation measurements.

8 Appendix A

In this section we prove how to transform the exponential distribution into a binary uniform distribution. Let P(0) denote the probability that the next bit extracted from the bitstream is 0 and P(1) analogous.

Lemma 1: $P(0) + P(1) = 1$

proof : According to the additivity of the Riemann-Integral we divide the cumulative distribution $P(0 \leq X \leq \infty)$ into 2 sums of integrals representing each summand either an even or an odd timing unit Δt. Note that the first sum of integrals represents $P(0)$ and the second sum $P(1)$.

$$\begin{aligned}
\int_0^\infty \lambda \cdot e^{-\lambda x} dx &= \sum_{N=0}^{\infty} \int_{(2N)\cdot\Delta t}^{(2N+1)\cdot\Delta t} \lambda \cdot e^{-\lambda \cdot s}\, ds \;+\; \sum_{N=0}^{\infty} \int_{(2N+1)\cdot\Delta t}^{(2N+2)\cdot\Delta t} \lambda \cdot e^{-\lambda \cdot s}\, ds \\
&= \sum_{N=0}^{\infty} \lambda \cdot \left[-\frac{1}{\lambda} \cdot e^{-\lambda \cdot s} \right]_{(2N)\cdot\Delta t}^{(2N+1)\cdot\Delta t} \;+\; \sum_{N=0}^{\infty} \lambda \cdot \left[-\frac{1}{\lambda} \cdot e^{-\lambda \cdot s} \right]_{(2N+1)\cdot\Delta t}^{(2N+2)\cdot\Delta t} \\
&= \sum_{N=0}^{\infty} \left(e^{-\lambda(2N)\cdot\Delta t} \;-\; e^{-\lambda(2N+1)\cdot\Delta t} \;+\; e^{-\lambda(2N+1)\cdot\Delta t} \;-\; e^{-\lambda(2N+2)\cdot\Delta t} \right) \\
&= \sum_{N=0}^{\infty} e^{-\lambda(2N)\cdot\Delta t} \;-\; e^{-\lambda(2N+2)\cdot\Delta t} \;=\; \sum_{N=0}^{\infty} e^{-\lambda(2N)\cdot\Delta t} \;-\; e^{-2\cdot\lambda(N+1)\cdot\Delta t}
\end{aligned}$$

The last series is a telescope sum that collapses to $e^{-\lambda 2 \cdot 0 \cdot \Delta t} = 1$. For all integers $N > 1$ consecutive summands add to zero. Hence, $P(0) + P(1) = 1$. □

The following two lemmas provide a discrete formula for both $P(0)$ and $P(1)$.

Lemma 2: $P(0) \;=\; \sum_{N=0}^{\infty} (-1)^N \cdot e^{-\lambda N \cdot \Delta t}$

proof : The sum representation of $P(0)$ in Lemma 1 delivers

$$\begin{aligned}
P(0) &= \sum_{N=0}^{\infty} \lambda \cdot \left[-\frac{1}{\lambda} \cdot e^{-\lambda s} \right]_{2N\cdot\Delta t}^{(2N+1)\cdot\Delta t} \\
&= \sum_{N=0}^{\infty} -\left(e^{-\lambda(2N+1)\cdot\Delta t} \;-\; e^{-\lambda(2N)\cdot\Delta t} \right) \;=\; \sum_{N=0}^{\infty} e^{-\lambda(2N)\cdot\Delta t} \;-\; e^{-\lambda(2N+1)\cdot\Delta t}
\end{aligned}$$

Expanding this sum produces:

$$e^{-0} \;-\; e^{-\lambda\cdot\Delta t} \;+\; e^{-\lambda 2\cdot\Delta t} \;-\; e^{-\lambda 3\cdot\Delta t} \;+\; \ldots$$

which leads to the following series:

$$\sum_{N=0}^{\infty} (-1)^N \cdot e^{-\lambda N \cdot \Delta t}$$

This series converges because $(-1)^N$ is alternating and $e^{-\lambda \cdot N \cdot \Delta t}$ is a monotone sequence leading to 0 (Leibnitz). $\square$

Lemma 3: $P(1) \;=\; \sum_{N=0}^{\infty}(-1)^N \cdot e^{-\lambda N \cdot \Delta t} \;\cdot\; e^{-\lambda \cdot \Delta t} \;=\; P(0) \cdot e^{-\lambda \cdot \Delta t}$

proof : Analogous to the previous case $P(0)$, we start with the sum representation as seen in Lemma 1

$$\begin{aligned} P(1) &= \sum_{N=0}^{\infty} \lambda \cdot \left[-\frac{1}{\lambda} \cdot e^{-\lambda s} \right]_{(2N+1)\cdot \Delta t}^{(2N+2)\cdot \Delta t} \\ &= \sum_{N=0}^{\infty} -\left(e^{-\lambda(2N+2)\cdot \Delta t} \;-\; e^{-\lambda(2N+1)\cdot \Delta t} \right) \;=\; \sum_{N=0}^{\infty} e^{-\lambda(2N+1)\cdot \Delta t} \;-\; e^{-\lambda(2N+2)\cdot \Delta t} \end{aligned}$$

An expansion of the sum leads to:

$$e^{-\lambda \cdot \Delta t} \;-\; e^{-\lambda 2 \cdot \Delta t} \;+\; e^{-\lambda 3 \cdot \Delta t} \;-\; e^{-\lambda 4 \cdot \Delta t} \;+\; \ldots$$

summing up to this series:

$$\sum_{N=0}^{\infty}(-1)^N \cdot e^{-\lambda(N+1)\cdot \Delta t}$$

The exponent $\lambda(N+1) \cdot \Delta t$ can be expanded to $-\lambda N \Delta t \;-\; \lambda \Delta t$ which is equivalent to

$$\sum_{N=0}^{\infty}(-1)^N \cdot e^{-\lambda N \cdot \Delta t} \;\cdot\; e^{-\lambda \cdot \Delta t} \;=\; P(0) \;\cdot\; e^{-\lambda \cdot \Delta t} \quad \square$$

Lemma 4: $P(1) - P(0) \longrightarrow 0$ for $\Delta t \longrightarrow 0$, *exponetially.*

proof :

$$\begin{aligned} P(1) - P(0) &= \left(\sum_{N=0}^{\infty}(-1)^N \cdot e^{-\lambda N \cdot \Delta t} \right) \cdot e^{-\lambda \cdot \Delta t} \;-\; \left(\sum_{N=0}^{\infty}(-1)^N \cdot e^{-\lambda N \cdot \Delta t} \right) \\ &= \sum_{N=0}^{\infty}(-1)^N \cdot e^{-\lambda N \cdot \Delta t} \;\cdot\; (e^{-\lambda \cdot \Delta t} - 1) \end{aligned}$$

Since $e^{-\lambda \cdot \Delta t} - 1$ converges to 0 and the series remains bounded, $P(1) - P(0)$ becomes negligible small if Δt approaches 0. $\square$

This leads to : $P(0) \approx P(1) \approx 0,5$ if Δt is suiable small. *q.e.d.*

9 Appendix B

The biasing of the equal distribution of an uncorrelated bit sequence can be achieved by the following algorithm, proposed by von Neumann [32]:
The bitstream is divided into groups of two consecutive bits (see example). These two bits can adopt 4 different pairs of values. Each pair delivers either an output bit (0 or 1) or no bit at all, according to the following function:

$$\begin{aligned} f \quad &: \quad \{0,1\}^2 \;\rightarrow\; \{0,1,-\} \\ f(x_1,x_2) \quad &:= \quad \begin{cases} 0 \;:\; (x_1,x_2) \;=\; (0,1) \\ 1 \;:\; (x_1,x_2) \;=\; (1,0) \\ - \;:\; \text{otherwise} \end{cases} \end{aligned}$$

Due to the uncorrelation property, the following probabilities are independent from the position i in the bitstream.

$$P(x_i = 0) := p$$
$$P(x_i = 1) := q$$

In this binary case it holds $q = 1-p$ such that $P(0,1) = (p \cdot (1-p)$ and $P(1,0) = (1-p) \cdot p$. Hence, the combination $(0,1)$ has the same probability to be generated than $(1,0)$. The resulting bitstream is uniformly distributed under the assumption that the pairs $(0,0)$ and $(1,1)$ deliver no output bit. The expected input is 8 bits to gain 2 bits which decreases the former generation rate at least by factor 4. □

Example:

biased stream	0 1	1 1	0 1	1 0	0 1	1 0	0 0	1 0
unbiased stream	0	-	0	1	0	1	-	1

10 Appendix C

10.1 The Microcontroller Source Code

```
;
; ------------------------------------------------------------------------
; ------------------------------------------------------------------------
;
; RANDy - PIC assembler listing
;
; Each decay event causes a 7 micro seconds HI-impulse on input pin
; RA3 and the PIC-controller polls for it. The random bit is extracted
; from the lsb of TMR0 at this time. TMR0 runs at system clock
; (18.432 MHz / 4). After capturing the random bit TMR0 is
; reset to 0 to guarantee the same conditions for each measurement.
; 8 random bits are wrapped into a byte and transferred via RS232
; @ 56,7 KB/s. The internal UART is used for this transmission.
;
; ------------------------------------------------------------------------
; ------------------------------------------------------------------------
;
                LIST P=16F628

                #include "P16F628.INC"

                __config _HS_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF

; Registers
savew1          equ     h'20'
savew2          equ     h'a0'
savest          equ     h'21'
wc1             equ     h'22'
wc2             equ     h'23'
wc3             equ     h'24'
bit_count       equ     h'25'
rnd_value       equ     h'26'
flag_reg        equ     h'27'
tmp             equ     h'28'
```

```
savepc          equ     h'29'

; Bits
SIGNAL          equ     3
TX_LED          equ     0
LOW_BATT_LED    equ     1
NO_PULSES_LED   equ     2


        ORG     0x000                   ; Programs starts at 0x0000

        goto    init

        org     h'0004'                 ; ISR starts at 0x0004

;
;--------------------------------
; INTERRUPT SERVICE ROUTINE (ISR)
;--------------------------------
;
; save registers

        movwf   savew1                  ; save W-Register
        movf    STATUS, W               ;
        clrf    STATUS                  ; ensure Bank 0
        movwf   savest                  ; save STATUS register
        movf    PCLATH, W               ;
        movwf   savepc                  ; save PClath

; now check interrupt source

        btfss   INTCON, RBIF            ; change on RB4..7
        goto    intend                  ; no, finish ISR

; determine reason for interrupt

        movf    PORTB, W
        andlw   b'10100000'             ; isolate RB5 and RB7
        movwf   tmp
        rlf     tmp, F
        btfss   STATUS, C               ; test RB7
        goto    no_pulses               ; RB7 is cleared -> no pulses!

        rlf     tmp, F
        rlf     tmp, F
        btfss   STATUS, C               ; test RB5
        goto    low_batt                ; RB5 is cleared -> low batt!

; write back all registers

intend  movf    savepc, W               ;
        movwf   PCLATH                  ; restore again PClath

        movf    savest, W               ;
        movwf   STATUS                  ; restore STATUS

        swapf   savew1, F               ; restore W without
```

```
        swapf   savew1, W              ; affecting STATUS

        retfie                         ; finish ISR



;
; -------------------------
; INITIALIZE THE WHOLE CHIP
; -------------------------
;

; turn off comparators

init    movlw   b'00000111'            ; Comparators Off
        movwf   CMCON

; initialize IO-Ports

        clrf    PORTA                  ; reset PORTA outputs

        movlw   b'00000100'            ; RB2(TX - Pin) = 1
        movwf   PORTB                  ;

        bsf     STATUS, RP0            ; RAM PAGE 1

        movlw   b'00001000'            ; portA all pins output,
        movwf   TRISA                  ; except RA3

        movlw   b'10100010'            ; RB6-RB4 and RB1(RX)=in,
        movwf   TRISB                  ; others out

; set baud rate to communicate with PC
; baud rate = 57600, no parity, 1 stop bit
; SPBRG = FOSC / (16 * baud rate) - 1
;       = 18432000 / (16 * 57600) - 1 = 19

        movlw   d'19'
        movwf   SPBRG
        movlw   b'00100100'            ; BRGH = high (2)
        movwf   TXSTA                  ; enable Async Transmission

        bcf     STATUS, RP0            ; RAM PAGE 0

        movlw   b'10010000'            ; enable Async Reception
        movwf   RCSTA

; init timer 0
; tmr0 is enabled and runs with clock/4 without prescaler
; T0CS = 0 (fosc/4), PSA = 1 (no prescaler)

        bsf     STATUS, RP0            ; RAM PAGE 1
        movlw   b'11011000'            ;
        movwf   OPTION_REG             ; configurations for TMR0
        bcf     STATUS, RP0            ; RAM Page 0

; provide a settling time for start up
```

```
        call    wait1s

; check, if one of the fail conditions is fullfilled right at the beginning
; ('low battery': RB5=0  or  'no pulses': RB7=0)

        movf    PORTB, W
        andlw   b'10100000'             ; isolate RB5 and RB7
        movwf   tmp
        rlf     tmp, F
        btfss   STATUS, C               ; Test RB 7
        goto    no_pulses               ; RB7 is cleared -> no pulses!

        rlf     tmp, F
        rlf     tmp, F
        btfss   STATUS, C               ; Test RB5
        goto    low_batt                ; RB5 is cleared -> low batt!

; enable interrupt surveillance of LOW BATT and NO PULSES

        clrf    INTCON                  ; clear all interrupt sources
        bsf     INTCON, RBIE            ; enable interrupt on RB4..7 change
        bsf     INTCON, GIE             ; enable global interrupt

;
; ---------
; MAIN LOOP
; ---------
;

loop    btfss   PORTA, SIGNAL           ; poll for SIGNAL to become HI
        goto    loop

        movf    TMR0, W                 ; isolate value from TMR0 in W
        clrf    TMR0                    ; reset TMR0
        andlw   b'00000001'             ; isolate LSB from timer value
        btfss   STATUS, Z
        goto    l1a
        bcf     STATUS, C
        goto    l1b

l1a     bsf     STATUS, C
l1b     rlf     rnd_value, F            ; shift random bit into rnd_value

l1c     btfsc   PORTA, SIGNAL           ; poll for Signal LO
        goto    l1c

;---

l2a     btfss   PORTA, SIGNAL           ; ... for bit 2
        goto    l2a

        movf    TMR0, W
        clrf    TMR0
        andlw   b'00000001'
        btfss   STATUS, Z
        goto    l2b
```

```
        bcf     STATUS, C
        goto    l2c

l2b     bsf     STATUS, C
l2c     rlf     rnd_value, F

l2d     btfsc   PORTA, SIGNAL
        goto    l2d

;---

l3a     btfss   PORTA, SIGNAL           ; ... for bit 3
        goto    l3a

        movf    TMR0, W
        clrf    TMR0
        andlw   b'00000001'
        btfss   STATUS, Z
        goto    l3b
        bcf     STATUS, C
        goto    l3c

l3b     bsf     STATUS, C
l3c     rlf     rnd_value, F

l3d     btfsc   PORTA, SIGNAL
        goto    l3d

;---

l4a     btfss   PORTA, SIGNAL           ; ... for bit 4
        goto    l4a

        movf    TMR0, W
        clrf    TMR0
        andlw   b'00000001'
        btfss   STATUS, Z
        goto    l4b
        bcf     STATUS, C
        goto    l4c

l4b     bsf     STATUS, C
l4c     rlf     rnd_value, F

l4d     btfsc   PORTA, SIGNAL
        goto    l4d

;---

l5a     btfss   PORTA, SIGNAL           ; ... for bit 5
        goto    l5a

        movf    TMR0, W
        clrf    TMR0
        andlw   b'00000001'
        btfss   STATUS, Z
        goto    l5b
```

```
        bcf     STATUS, C
        goto    l5c

l5b     bsf     STATUS, C
l5c     rlf     rnd_value, F

l5d     btfsc   PORTA, SIGNAL
        goto    l5d

;---

l6a     btfss   PORTA, SIGNAL           ; ... for bit 6
        goto    l6a

        movf    TMR0, W
        clrf    TMR0
        andlw   b'00000001'
        btfss   STATUS, Z
        goto    l6b
        bcf     STATUS, C
        goto    l6c

l6b     bsf     STATUS, C
l6c     rlf     rnd_value, F

l6d     btfsc   PORTA, SIGNAL
        goto    l6d

;---

l7a     btfss   PORTA, SIGNAL           ; ... for bit 7
        goto    l7a

        movf    TMR0, W
        clrf    TMR0
        andlw   b'00000001'
        btfss   STATUS, Z
        goto    l7b
        bcf     STATUS, C
        goto    l7c

l7b     bsf     STATUS, C
l7c     rlf     rnd_value, F

l7d     btfsc   PORTA, SIGNAL
        goto    l7d

;---

l8a     btfss   PORTA, SIGNAL           ; ... finally for bit 8
        goto    l8a

        movf    TMR0, W
        clrf    TMR0
        andlw   b'00000001'
        btfss   STATUS, Z
        goto    l8b
```

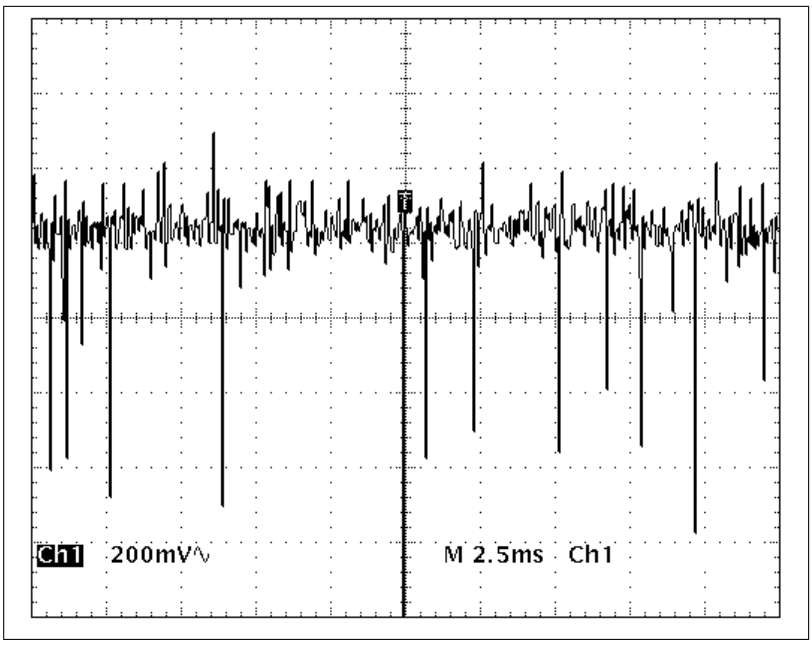
```
        bcf     STATUS, C
        goto    l8c

l8b     bsf     STATUS, C
l8c     rlf     rnd_value, F

; rnd_value is now complete, send it!

        movf    PORTA, W                ; blink TX_LED
        xorlw   b'00000001'
        movwf   PORTA

        movf    rnd_value, W            ; move rnd_value to W

; send character in W via RS232 and wait until finished sending

        movwf   TXREG                   ; send data in W
        bsf     STATUS, RP0             ; RAM PAGE 1
swait   btfss   TXSTA, TRMT             ; wait for transmission
        goto    swait                   ; to be completed
        bcf     STATUS, RP0             ; RAM PAGE 0

; resynchronize
; reason: we might have skipped 1 or more impulses during transmission
; of the random byte

l8d     btfsc   PORTA, SIGNAL           ; wait for signal to become LO
        goto    l8d

l9      btfss   PORTA, SIGNAL           ; wait for signal to become HI
        goto    l9

        nop                             ; skip 1 instr., no eval. of TMR0
        clrf    TMR0                    ; reset TMR0

l10     btfsc   PORTA, SIGNAL           ; poll for SIGNAL to become LO
        goto    l10

        goto    loop                    ; jump back to main loop


;
; --------------------
; WAIT ABOUT 1 SECOND
; --------------------
;

wait1s  movlw   d'10'                   ; 10 * 0.1s = 1s
        movwf   wc3

wl3     movlw   d'200'                  ; 200*500s = 100ms = 0.1s
        movwf   wc2

wl2     movlw   d'200'                  ; 200*2,5s = 500 s
        movwf   wc1

wl1     nop                             ; 1
```

```
        nop                             ; 2
        nop                             ; 3
        nop                             ; 4
        nop                             ; 5
        nop                             ; 6
        nop                             ; 7
        decfsz  wc1, F                  ; 8
        goto    wl1                     ; 9+10 = 2,5s

        decfsz  wc2, F
        goto    wl2

        decfsz  wc3, F
        goto    wl3

        return


;
; ------------------------------------
; DEAD END WHEN 'LOW BATT' IS INDICATED
; ------------------------------------
;
low_batt:
        bcf     PORTA, TX_LED           ; clear TX_LED in every case
        bsf     PORTA, LOW_BATT_LED     ; indicate low batt
lb1     goto    lb1                     ; stay here forever


;
; -------------------------------------
; DEAD END WHEN 'NO PULSES' IS INDICATED
; -------------------------------------
;
no_pulses:
        bcf     PORTA, TX_LED           ; clear TX_LED in every case
        bsf     PORTA, NO_PULSES_LED    ; indicate no pulses
np1     goto    np1                     ; stay here forever


        END
```

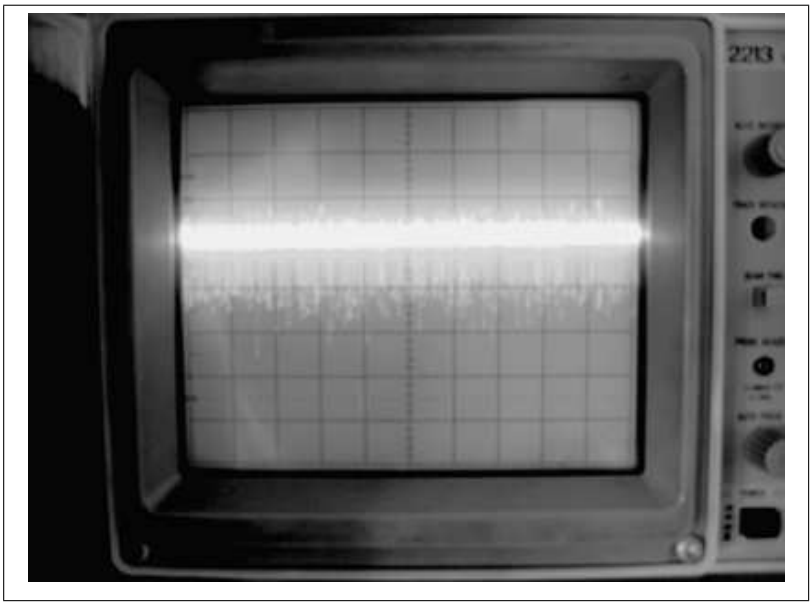
10.2 The Compiled HEX-file To Transfer To The Micro

```
:020000040000FA
:020000001F28B7
:08000800A00003088301A10020
:100010000A08A9000B1C18280608A039A800A80E79
:10002000A80CA80C031CD228A80CA80C031CD528CB
:1000300029088A0021088300A00E200E090007303D
:100040009F00850104308600831608308500A230A9
:10005000860013309900243098008312903098006 5
:100060008316D83081008312BE200608A039A8006C
:10007000A80EA80CA80C031CD228A80CA80C031CC2
:10008000D5288B018B158B17851D4428010881010C
:100090000139031D4D2803104E280314A60D8519A0
:1000A0004F28851D5128010881010139031D5A2857
:1000B00003105B280314A60D85195C28851D5E2896
:1000C000010881010139031D672803106828031402
:1000D000A60D85196928851D6B2801088101013944
:1000E000031D7428031075280314A60D851976289E
:1000F000851D782801088101013903 1D812803101D
:1001000082280314A60D85198328851D85280108DA
:10011000810101390 31D8E2803108F280314A60DB9
:10012000851990288 51D92280108810101390 31D38
:100130009B2803109C280314A60D85199D28851D56
:100140009F28010881010139031DA8280310A9284F
:100150000314A60D0508013A85002608B820851964
:10016000AF28851DB128000081018519B5284428D4
:1001700099008316981CBA28831208000A30A4003C
:10018000C830A300C830A2000000000000000003A
:1001900000000000000A20BC428A30BC228A40B7F
:1001A000C028080005108514D42805100515D72887
:02400E00E23F8F
:00000001FF
```

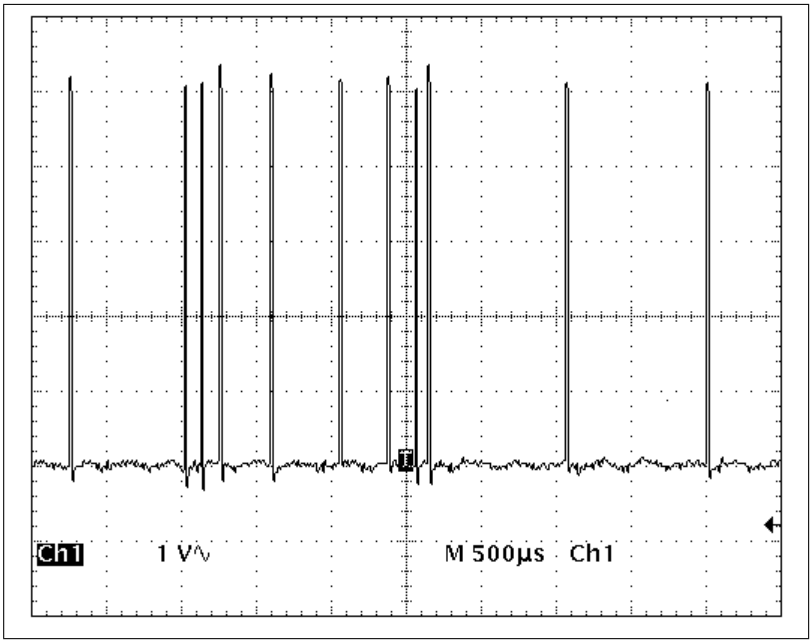
10.3 Some oscilloscope screenshots



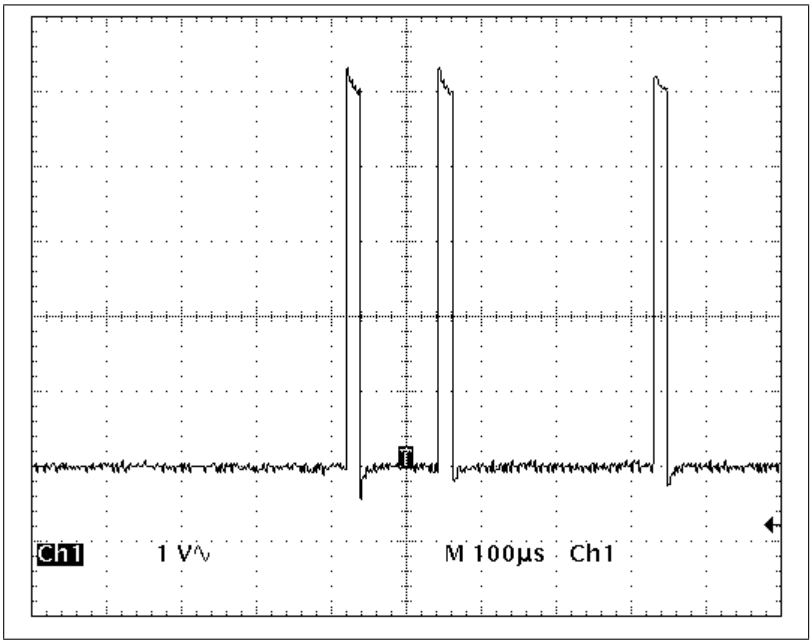
The signal injected into the comparator. The spikes are caused by the decay events. It is obvious that not every alpha particle has the same energy since they have no equal peak. But only the timing aspect is interesting so the comparator outputs the short voltage breakdowns below a certain level.



And the same view from an analog oscilloscope. Unfortunatelly there are several beam plots from left to right accumulated in one picture so it almost looks like white noise.

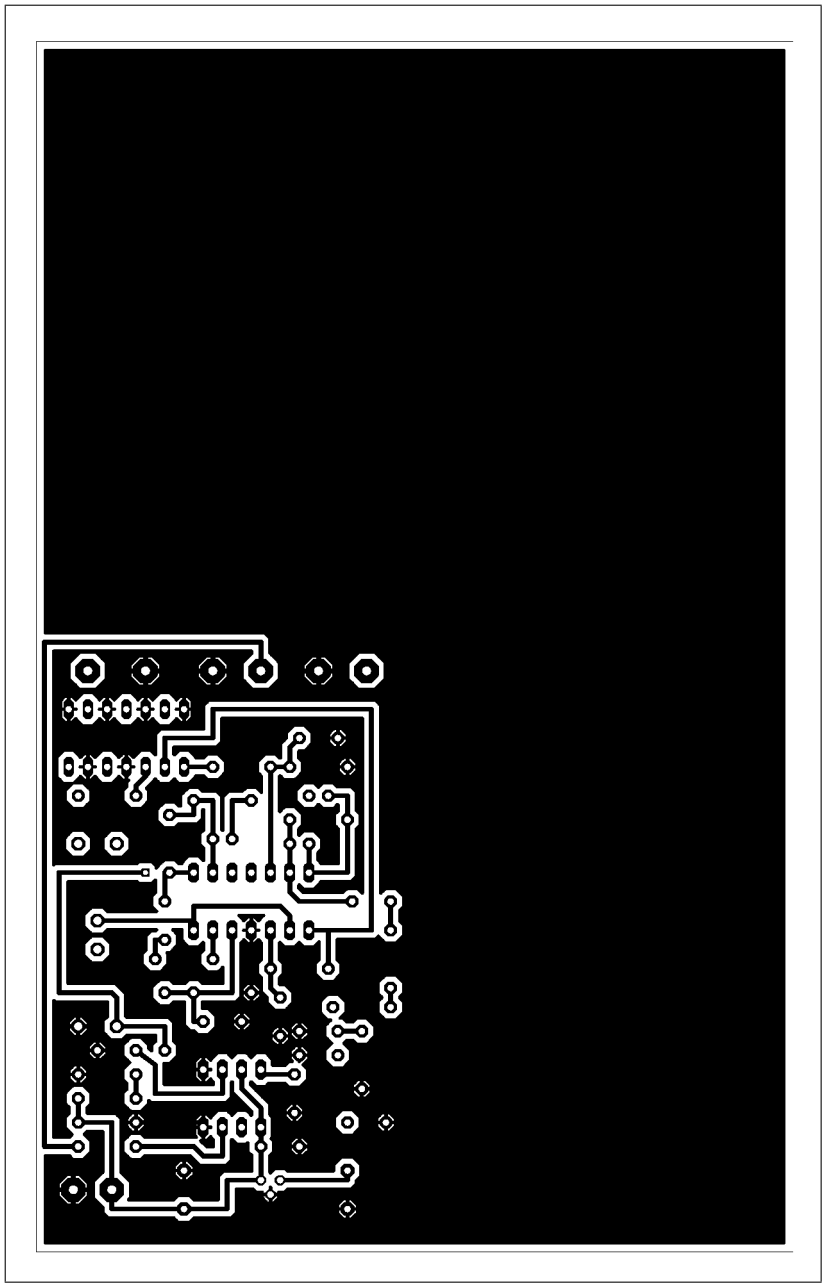


The spikes in the digital section behind the monoflop. All impulses have the same amplitude and the same length and are ready for processing in the microcontroller.

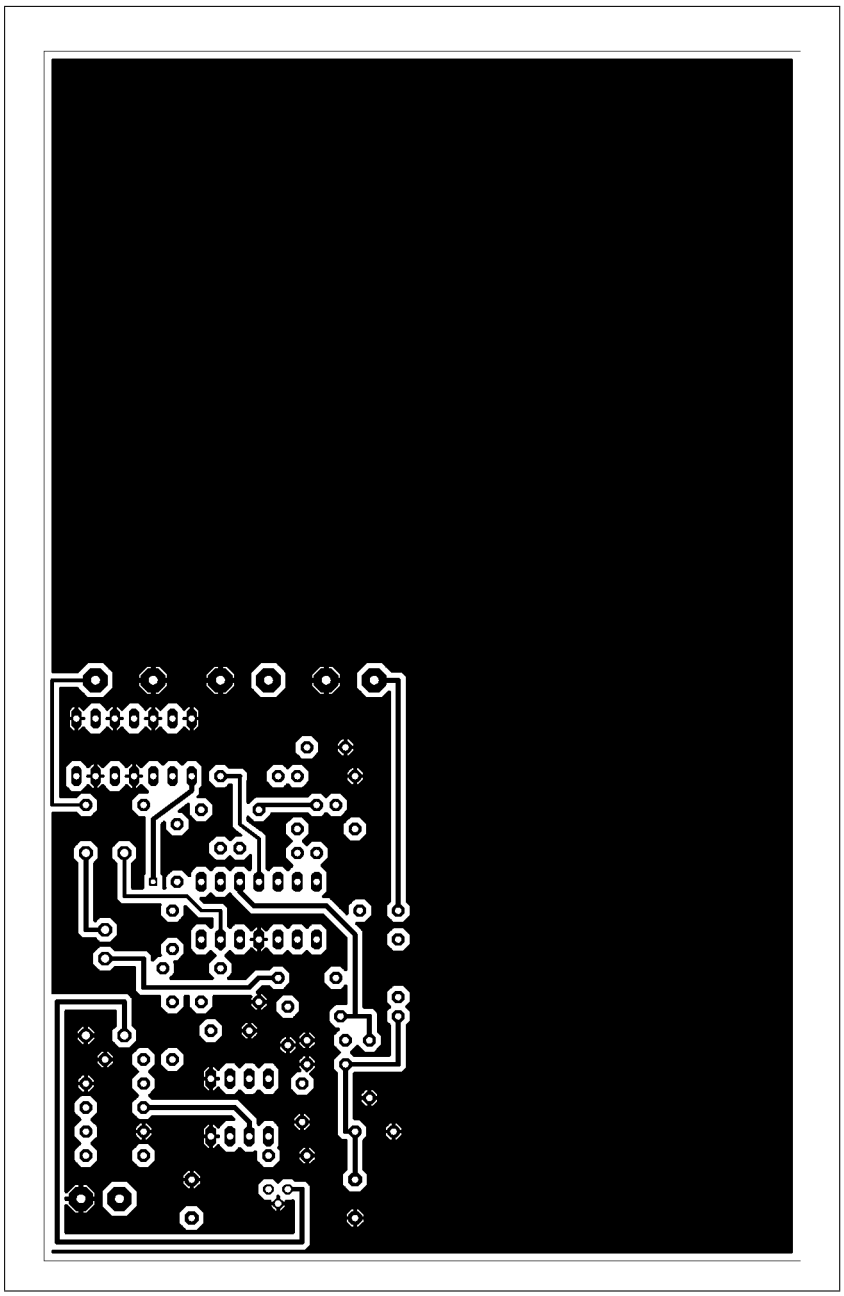


A more detailed look on the uniformed pulses in the digital section.

10.4 The PCB of the Analog Section

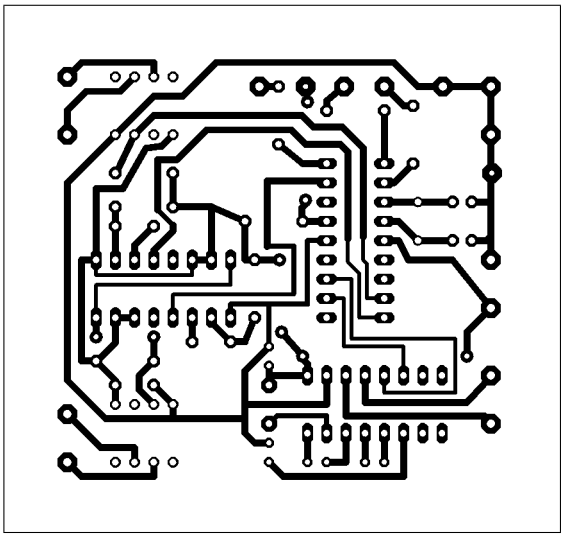


Layout of the bottom side

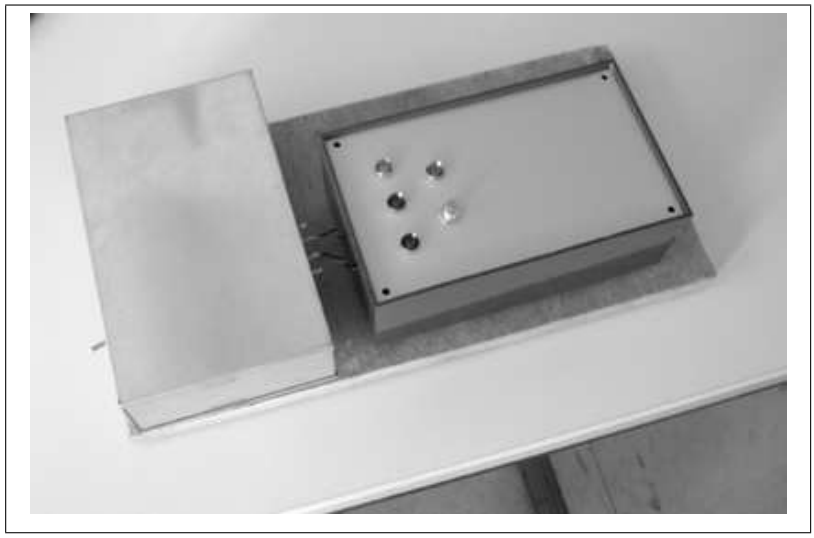


Layout of the upper side

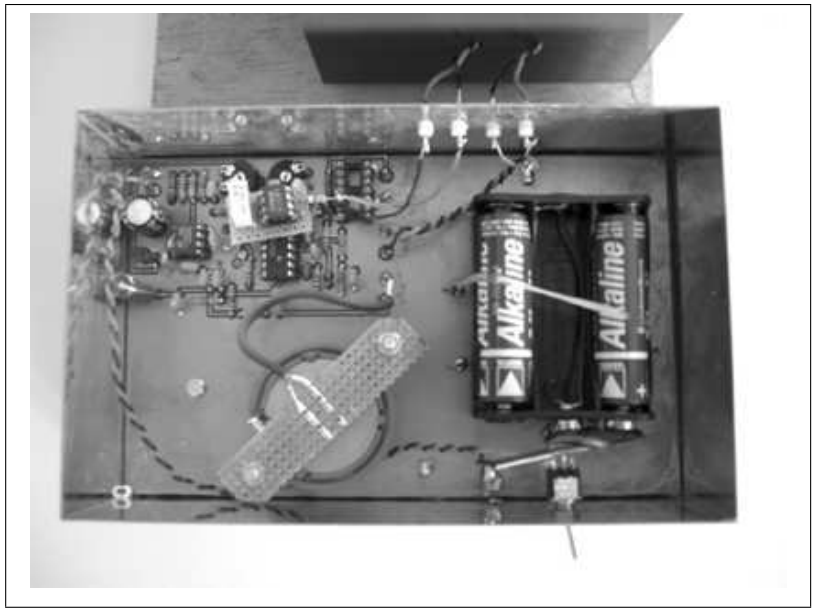
10.5 The PCB of the Digital Section



10.6 Some Photos



The whole random generator: The pulse detector in the metal case on the left and the digital section with the power supply and several control LEDs and the reset button on the right.



A detailed view of the analog section with the radiation source, the mounted photo diode and the daughter board soldered above the main board. The battery pack is fixed with some rubber bands.

The digital part with the microcontroller and the power supply.

References

[1] Ammar Alkassar. Zufallszahlengenerator, 1998.

[2] Hans Breuer. *dtv-Atlas zur Chemie, Band 1.* Deutscher Taschenbuch Verlag, 1992.

[3] Marshall Brian. Inside a smoke detector. available at : `http://science.howstuffworks.com/inside-smoke.htm`.

[4] Centrovision. A primer on photodiode technology. `http://www.centrovision.com/tech2.html`.

[5] Westphal Electronic. Zrandom. `http://home.t-online.de/home/p.westphal/zran_eng.htm`, 2003. commercial device.

[6] Aleksandr Figotin, Ilya Vitebskiy, Vadim Popovich, Gennady Stetsenko, Stanislav Molchanov, Alexander Gordon, Joseph Quinn, and Nicholas Stavrakas. Random number generator based on the spotaneous alpha-decay. *U.S. patent Appl. No.: 10/127,221*, 2003.

[7] Gill and others. Radiation damage studies if optical link components for applications in future high energy physics experiments. available at : `http://www.cern.ch/CERN/Divisions/ECP/CME/OpticalLinks/wdocs/spiepap.pdf`.

[8] I. Goldberg and D. Wagner. Randomness in the netscape browsers. *Dr. Dobb's Journal*, January 1996.

[9] Nigel Goodwin. Winpicprog. available at : `http://www.winpicprog.co.uk`.

[10] Jerald Graeme. *Photodiode Amplifiers - OP amp solutions.* McGraw-Hill, 1996.

[11] Michael Gude. Concept for a high performance random number generator based on physical random phenomena. *Frequenz*, 39:187–190, 7/8 1985.

[12] Michael Gude. *Ein quasi-idealer Gleichverteilungsgenerator basierend auf physikalischen Zufallsphänomenen.* PhD thesis, RWTH Aachen, 1987.

[13] Grzegorz Hahn. Photodiode and op amps form wideband radiation monitor. *Electronics Designer's Casebook*, (3), 1978.

[14] Frank A. Haight. *Handbook of the Poisson Distribution.* John Wiley & Sons, Inc., 1967.

[15] Texas Instruments. Datasheet : TTL, TLC274, TLC7702. `http://www.ti.com`.

[16] Benjamin Jun and Paul Kocher. The intel number generator. White paper prepared for intel corporation, Cryptography Research, Inc., April 1999.

[17] Günther Kraus. Stochastische abhängigkeit von in schneller folge erzeugten gleichverteileten diskreten zufallsereignissen. *Frequenz*, 35:274–277, 10 1981.

[18] George Marsaglia. Diehard: a battery of tests for random number generators. `http://stat.fsu.edu/~geo/diehard.html`.

[19] Microchip. Microchip mp-lab, datasheet : PIC16F628. `http://www.microchip.com`.

[20] Herschell F. Murry. A general approach for generating natural random variables. *IEEE Transactions on Computers*, pages 1210–1214, 1970.

[21] National Institute of Standards and Technology. Random number generation and testing. `http://csrc.nist.gov/rng/`.

[22] Maxim Integrated Products. Datasheet : MAX232. `http://www.maxim-ic.com`.

[23] Manfred Richter. *Ein Rauschgenerator zur Gewinnung von quasi-idealen Zufallszahlen für die stochastische Simulation.* PhD thesis, RWTH Aachen, Shaker-Verlag, 1992.

[24] Mike Rosing and Patrick Emin. Ionization from alpha decay for random bit generation. available at : `http://www.terracom.net/~Eeresrch/detecting_random.ps`, 1999.

[25] Fairchild Semiconductor. Datasheet : 6N137, 6N139. `http://www.fairchildsemi.com`.

[26] National Semiconductor. Datasheet : LM311, LM 2936-5.0. `http://www.national.com`.

[27] Osram Opto Semiconductors. Datasheet : BPX61. `http://www.osram-os.com`.

[28] Andre Stefanov, Nicolas Gisin, Olivier Guinnard, Laurent Guinnard, and Hugo Zbinden. Optical quantum random number generator. *online*, 1999.

[29] W. E. Thomson. Ernie - a mathematical and statistical analysis. *Journal of the Royal Statistical Society, Series B*, 101:301–333, 1959.

[30] C. H. Vincent. The generation of truely random binary numbers. *Journal of Physics E: Scientific Instruments*, 3:594–598, 1970.

[31] C. H. Vincent. Precautions for accuracy in the generation of truely random binary numbers. *Journal of Physics E: Scientific Instruments*, 4:825–828, 1971.

[32] John von Neumann. Various techniques used in connection with random digits. *Applied Mathematics Series*, (12):36–38, 1951.

[33] John Walker. ent - a pseudorandom sequence test program. `http://www.fourmilab.ch/random/`.

[34] John Walker. Hotbits. online : `http://www.fourmilab.ch/hotbits/`, 2003.

[35] Charles Wenzel. Fun with ion chambers. available at : `http://www.techlib.com/science/ion.html`, 2003.