# Obtenção de sequências aleatórias a partir de interferência eletromagnetica

**Cristiano M. Matsui[1], Kallil M. Caparroz[2], Rodrigo C. Anater[3]**

[1]Universidade Tecnológica Federal do Paraná - Câmpus Pato Branco

cristiano.matsui@gmail.com, kallil@alunos.utfpr.edu.br, rodrigoanater@alunos.utfpr.edu.br

***Abstract.*** *The definition of what is meant by a random number or event is the subject of discussion by professionals in many fields of study, including computing professionals. However, most part of the computers used in the present days generate only pseudorandom numbers. This article details the development of a random number generator using the electrical noise obtained from the analog port of a microcontroller*

***Resumo.*** *A definição do que se entende por um número ou evento aleatório é alvo de debates por parte de profissionais de diversas áreas de estudo, inclusive profissionais de computação [Volchan 2001]. Entretanto, a maioria dos computadores presentes na atualidade geram apenas números pseudo-aleatórios. O presente artigo detalha o desenvolvimento de um gerador de números aleatórios utilizando o ruído elétrico obtido da porta analógica de um microconrolador.*

## 1. Introdução

Um número aleatório é um elemento de uma sequência numérica que apresenta aleatoriedade estatística, ou seja, uma sequencia que não apresente nenhuma regularidade ou padrão, de forma que o valor de um elemento qualquer não possa ser previsto a partir dos valores prévios na sequência.

Na área computacional, a utilização de números aleatórios é de grande utilidade em diversas situações, como em simulações de situações reais, que contenha fatores externos aos controlados, sistemas de jogos de apostas, que devem ter resultados imprevisíveis à qualquer pessoa, e no contexto de segurança, na geração de chaves criptográficas. Porém, sem uma fonte de dados aleatórios, um computador não é capaz de gerar sequências verdadeiramente aleatórias, e acaba se baseando em sistemas de gerações pseudoaleatórias. Este projeto objetiva a criação de um sistema capaz de gerar sequências numéricas verdadeiramente aleatórias, a partir da interferência (ruído) eletromagnética captada por um microcontrolador.

## 2. Geração de Sequências Aleatórias

Em geral, sistemas computacionais são determinísticos, ou seja, cada evento é definido pela sequência de eventos precedentes. Essa é basicamente a definição contrária à definição de aleatoriedade, criando assim um problema na geração de números aleatórios por computadores. Porém, existem duas maneiras principais de se resolver esse problema. Uma forma é a utilização de sequências pseudoaleatórias. Essas sequências se baseiam em uma série de operações determinísticas que, a partir

de um conjunto de parâmetros, são capazes de gerar sequências que se pareçam suficientemente aleatórias para alguma finalidade específica. Esse método tem algumas vantagens, como uma grande eficiência, gerando grandes sequências em um curto período de tempo, e reprodutibilidade, que permite a repetição de testes, uma vez que o mesmo conjunto de parâmetros iniciais gera a mesma sequência.

Porém, em muitos casos, como para criptografia, a reprodutibilidade dos dados não é desejada, pois os valores devem ser realmente imprevisíveis. Para tal, existem os chamados geradores de números verdadeiramente aleatórios, que devido à natureza determinística de sistemas computacionais, necessitam de obtenção de dados externos ao computador.

É comum geradores pseudoaleatórios se utilizarem de algum dado externo como parâmetro para a geração de suas sequências, como o tempo do sistema, de forma à variar seus resultados. Já geradores de sequências numéricas verdadeiramente aleatórios, dependem de uma fonte de dados aleatórios para a geração da sequência, dados estes que devem apresentar alguma forma de aleatoriedade intrínseca. Assim, geradores de números verdadeiramente aleatórios costumam se utilizar da medição de fenômenos físicos, como sons atmosféricos ou até mesmo decaimento radioativo.

# 3. Materiais e Métodos

## 3.1. Materiais

Para captar e realizar a leitura do ruído eletromagnético foram utilizados uma antena para a captação e um microcontrolador para a aquisição do valor do ruído captado.

### 3.1.1. Arduino UNO

O Arduino Uno (Fig 1) é uma placa de microcontrolador baseado no ATmega328 (datasheet). Possui 14 pinos de entrada/saída digital (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal oscilador de 16MHz, uma conexão USB e uma entrada de alimentação uma conexão ICSP.

O microcontrolador foi responsável pela leitura do ruído eletromagnético em uma de suas portas analógicas. O ruído foi então convertido para um valor numérico através do conversor analógico-digital (ADC) presente no Arduino.

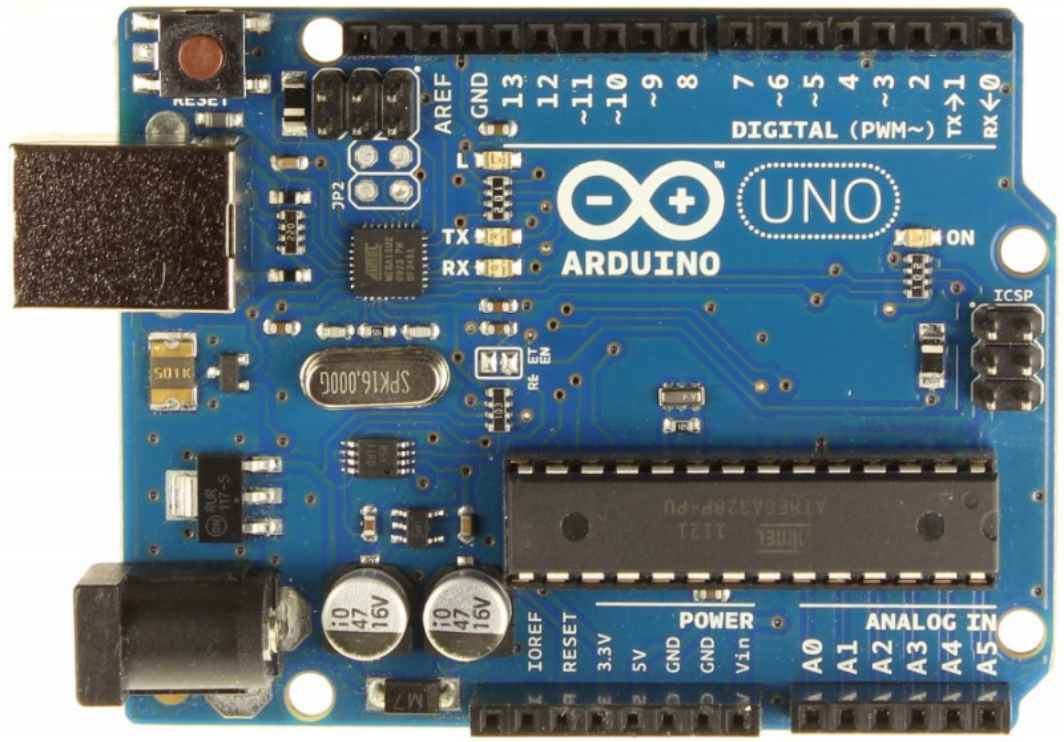


**Figura 1. Arduino UNO**

### 3.1.2. Antena

Foi utilizado um fio de estanho na porta do microcontrolador como aparato para a captação de interferência eletromagnética. A configuração escolhida da antena foi de monopolo de quarto de onda [Balanis 2005]. A frequência captada pela antena é prevista pela equação abaixo:

$$f = \frac{c}{\frac{L}{4}} \tag{1}$$

sendo $f$ a frequência em Hertz, $c$ a velocidade da luz e $L$ o comprimento da antena. Para uma antena de 8 centímetros, a frequência captada pela antena é de 15MHz.

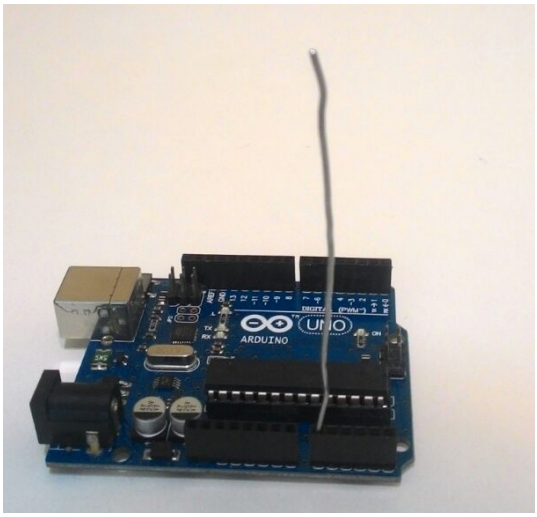


**Figura 2. Arduino UNO com a antena monopolo**

### 3.2. Métodos

Os dados adquiridos pela antena foram tratados no próprio microcontrolador. O ruído ocasionou uma pequena variação de tensão na porta analógica do Arduino UNO, que foram convertidos para valores na faixa de 0 até 1023. Foram realizadas dez mil leituras, com um pequeno delay (cerca de 20 milisegundos) entre leituras sucessivas, para evitar ler o mesmo ruído múltiplas vezes. O processamento destas leituras foi feito de acordo com o algoritmo abaixo.

**Algoritmo 1:** Algoritmo de tratamento de dados

**Input:** Ruído da porta analógica do microcontrolador
**Output:** Sequência de números em base hexadecimal
Início
**while** *existirem números a serem adquiridos* **do**
1.$prev\_value \leftarrow$ valor lido da porta do microcontrolador
2.$value \leftarrow$ próximo valor lido da porta do microcontrolador
3.$d \leftarrow |value - prev\_value|$
4.$d \leftarrow d \mod 16$
5.Imprima $d$ na base hexadecimal
**end**

Este código fornece uma sequência de números na base hexadecimal (0-F). Para as ferramentas de validação do experimento, era necessário um arquivo binário. Para isso, foi utilizado um código em C para ler os números em hexadecimal e transformá-los em uma sequência binária.

```
#include <stdio.h>

int main(void) {
   unsigned int a;
   while(scanf("%X", &a) != -1) {
```

```
      printf("%c%c", a>>8, a);
   }
   return 0;
}
```

**Código 1. Conversão hexa-binário**

## 4. Validação dos resultados

Para avaliar o nível de aleatoriedade das sequências, foi utilizado o programa "Ent", que foi criado com o objetivo de realizar a avaliação de geradores de sequências numéricas pseudoaleatórias para criptografia e aplicações de amostragem estatística. O programa realiza cinco testes diferentes:

- Entropia: Analisa a densidade de informação apresentada, ou seja, verifica o quanto a sequência pode ser comprimida e ainda representar o dado original. Quanto menor for a entropia, menos aleatória e mais compressível é a sequência.
- Teste Qui-quadrado: É calculada a distribuição qui-quadrado, que avalia a relação entre os valores fornecidos e os esperados. A distribuição é simétrica, sendo que sequências verdadeiramente aleatórias devem resultar em valores mais centralizados [Knuth 1998].
- Média Aritmética: Realiza a média aritmética dos valores. Considerando que a analise é feita a partir de blocos de quatro bytes, ou seja, são valores entre zero e 255, idealmente a média deve se aproximar de 127,5.
- Aproximação de $\pi$ pelo método de Monte Carlo: Cada sequência sucessiva de seis bytes é utilizada como coordenadas em um quadrado, o qual contem um circulo inscrito. Pontos verdadeiramente aleatórios tem uma probabilidade de estar dentro do circulo igual à razão entre a área do circulo pela área do quadrado. Assim, a razão entre os pontos do circulo e o total de pontos deve tender (lentamente) à um quarto do valor de $\pi$.
- Coeficiente de correlação serial: Mede o quanto cada elemento da sequência se relaciona ao elemento anterior, e para sequências verdadeiramente aleatórias, deve se aproximar de zero.

Deve-se ressaltar, porém, que não existem testes definitivos de aleatoriedade, uma vez que qualquer teste deve-se basear em probabilidades, e não certezas. Dessa forma, é de se esperar que, após realizada uma certa quantidade de testes, uma sequência verdadeiramente aleatória falhe em alguns deles, porém com menor probabilidade de que uma sequência não aleatória

## Referências


Balanis, C. A. (2005). *Antenna Theory: Analysis and Design*. Wiley-Interscience, New York, NY, USA.

Knuth, D. E. (1998). The art of computer programming, 2: seminumerical algorithms, addision wesley. *Reading, MA*.

Park, S. K. and Miller, K. W. (1988). Random number generators: good ones are hard to find. *Communications of the ACM*, 31(10):1192–1201.

Volchan, S. B. (2001). The algorithmic theory of randomness.