

Jogo de adivinhação de números aleatórios

Objetivo

Gerar números aleatórios com o Arduino e um display de sete segmentos. Abaixo está um exemplo simples de circuito para gerar números aleatórios entre 0 e 9 e exibi-los em um display de 7 segmentos comum (cátodo comum), utilizando o Arduino.

Montagem do circuito

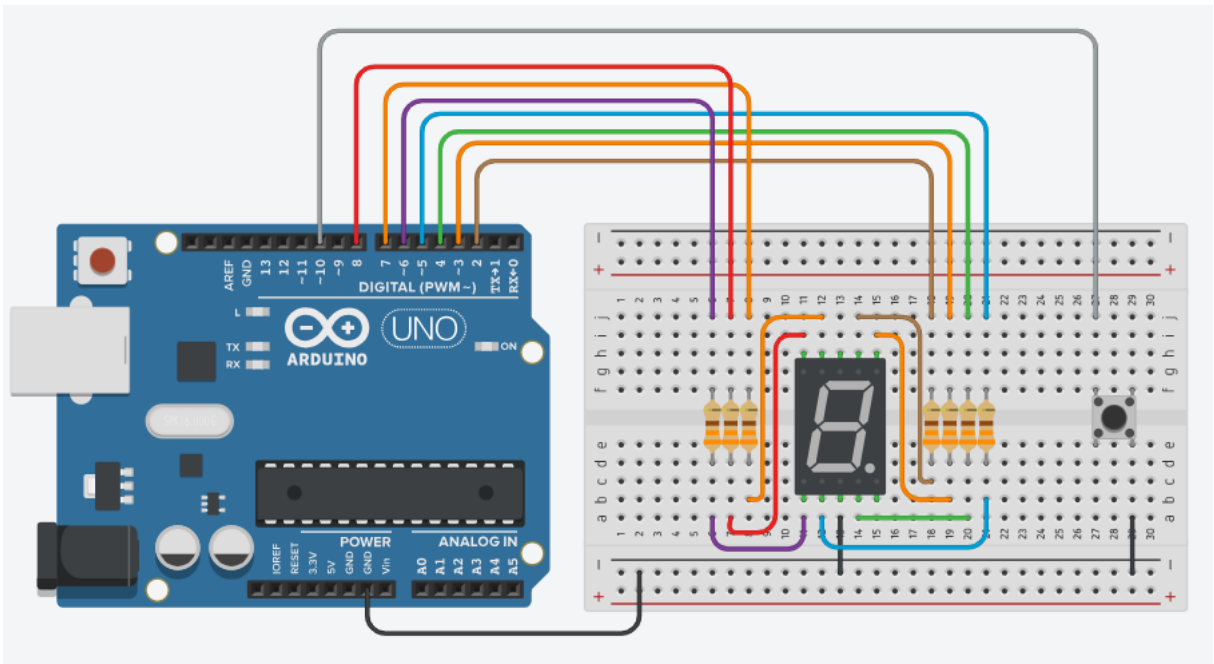


Figura 1 - Esquema de montagem do circuito.

Lista de materiais

1	Display de 7 segmentos (comum cátodo)
1	Arduino UNO
7	Resistores 330R (laranja, laranja, marrom)
1	Botão tipo tátil (push-button)
	Cabinhos e fios para ligações

Esquema de ligação do display

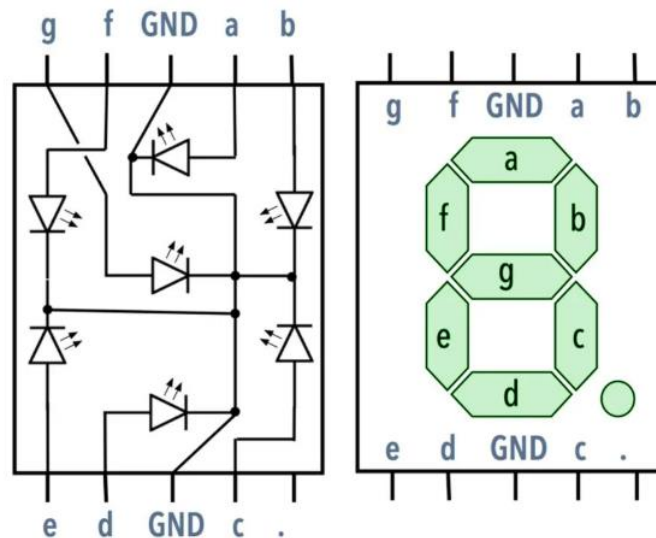


Figura 2 – Pinout de um display de 7 segmentos catodo comum. Fonte: <https://www.casadarobotica.com/display/display/7-segmentos/3x-display-7-segmento-1-digito-verde-catodo-comum-5161bg>

A tabela abaixo apresenta como devem ser as ligações de cada pino do display aos pinos do Arduino, mas observe pela figura da montagem do circuito que entre cada pino do display e cada pino do Arduino, existe um resistor de 330 Ohms.

Segmento	Pino do display	Pino do Arduino
A	a	2
B	b	3
C	c	4
D	d	5
E	e	6
F	f	7
G	g	8
COM	Cátodo comum	GND

Teste e uso

Ao energizar o Arduino, seja conectando o cabo USB ao computador ou usando bateria externa, o display irá apresentar uma contagem regressiva de 9 a 0. Ao chegar no zero, é só pressionar e segurar o botão pelo tempo que achar adequado para que os números sejam sorteados, ao soltar o display irá mostrar um número aleatório. Este projeto serve tanto para brincar de tentar adivinhar o número, como um dado eletrônico para jogos de tabuleiro.

Código de programação

NumerosAleatorios7seg.ino

```
1  /*****
2  * Projeto: Números aleatórios - Display 7 seg. *
3  * Desenvolvedor: Vanderlei Alves *
4  * Data: 27/05/2025 *
5  *****/
6  //Pinos ligados aos segmentos A-G
7  int segmentos[] = {2, 3, 4, 5, 6, 7, 8};
8
9  //Mapas dos dígitos de 0 a 9
10 byte digitos[10][7] = {
11   {1,1,1,1,1,0}, // 0
12   {0,1,1,0,0,0}, // 1
13   {1,1,0,1,1,0,1}, // 2
14   {1,1,1,1,0,0,1}, // 3
15   {0,1,1,0,0,1,1}, // 4
16   {1,0,1,1,0,1,1}, // 5
17   {1,0,1,1,1,1,1}, // 6
18   {1,1,1,0,0,0,0}, // 7
19   {1,1,1,1,1,1,1}, // 8
20   {1,1,1,1,0,1,1} // 9
21 };
22
23 //Variáveis globais
24 bool btnSortear = true;
25 int nOrder = 0;
26
27 void setup() {
28   //Inicialização da comunicação serial
29   Serial.begin(9600);
30
31   //Determinação do pino de entrada do botão
32   pinMode(10, INPUT_PULLUP);
33
34   //Definindo os pinos de 2 a 8 como saídas digitais
35   for (int i = 0; i < 7; i++) {
36     pinMode(segmentos[i], OUTPUT);
37   }
38
39   //Inicializa com valor aleatório usando o canal analógico A0 como gerador de aleatoriedade
40   randomSeed(analogRead(0));
41
42   //Realiza uma contagem inicial de 0 a 9
43   while(nOrder < 10)
44   {
45     mostrarNumero(nOrder);
46     Serial.println(nOrder);
47     nOrder++;
48     if(nOrder > 10) nOrder = 0;
49     delay(500);
50   }
51
52   //Zera o display
53   mostrarNumero(0);
54
55 }
```

```
56
57 void loop() {
58     //Leitura do botão
59     btnSortear = digitalRead(10);
60
61     //Verifica se o botão foi pressionado...
62     if(!btnSortear)
63     {
64         //...se sim, gera números aleatórios.
65         int numero = random(0, 10); // Número entre 0 e 9
66
67         //Apresenta o número gerado após o botão ser liberado
68         mostrarNumero(numero);
69         Serial.println(numero);
70     }
71 }
72
73 //Função que realiza a formação dos dígitos no display
74 void mostrarNumero(int n) {
75     for (int i = 0; i < 7; i++) {
76         digitalWrite(segmentos[i], digitos[n][i]);
77     }
78 }
```