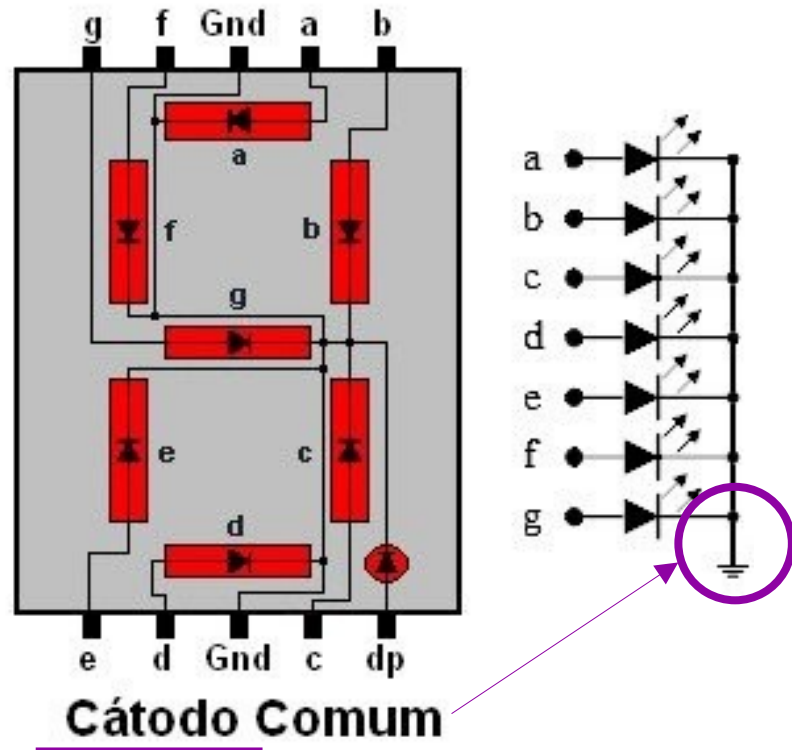
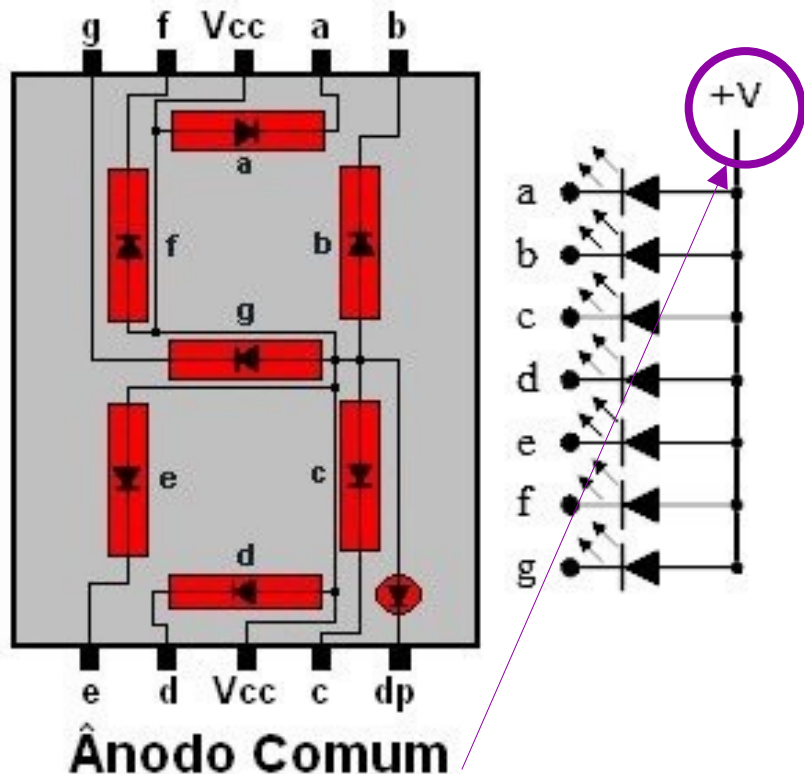


Display 7 segmentos

Displays de LED de 7 segmentos

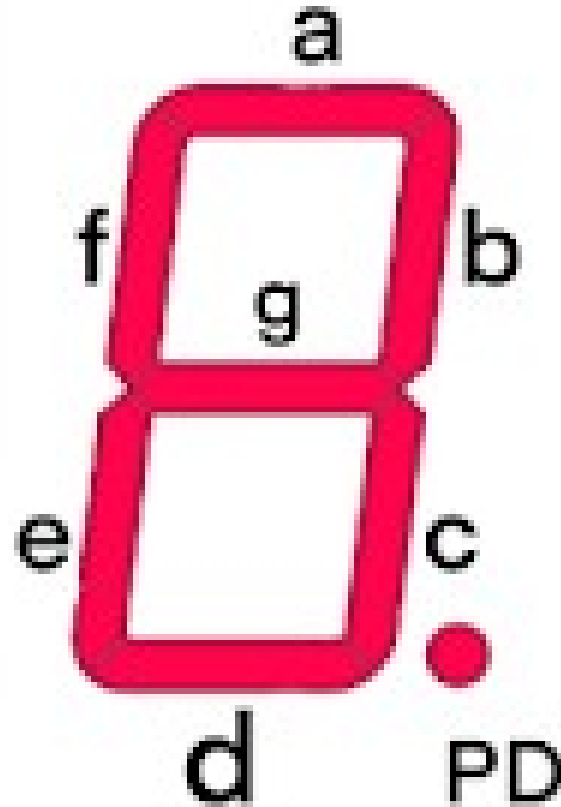
- São formados por LEDs dispostos em ‘segmentos’, os quais são identificados pelas letras de ‘a’ até ‘g’, mais um para o ponto decimal (PD, ou DP em Inglês)
- Podem ser do tipo ânodo comum ou cátodo comum

Displays de LED de 7 segmentos

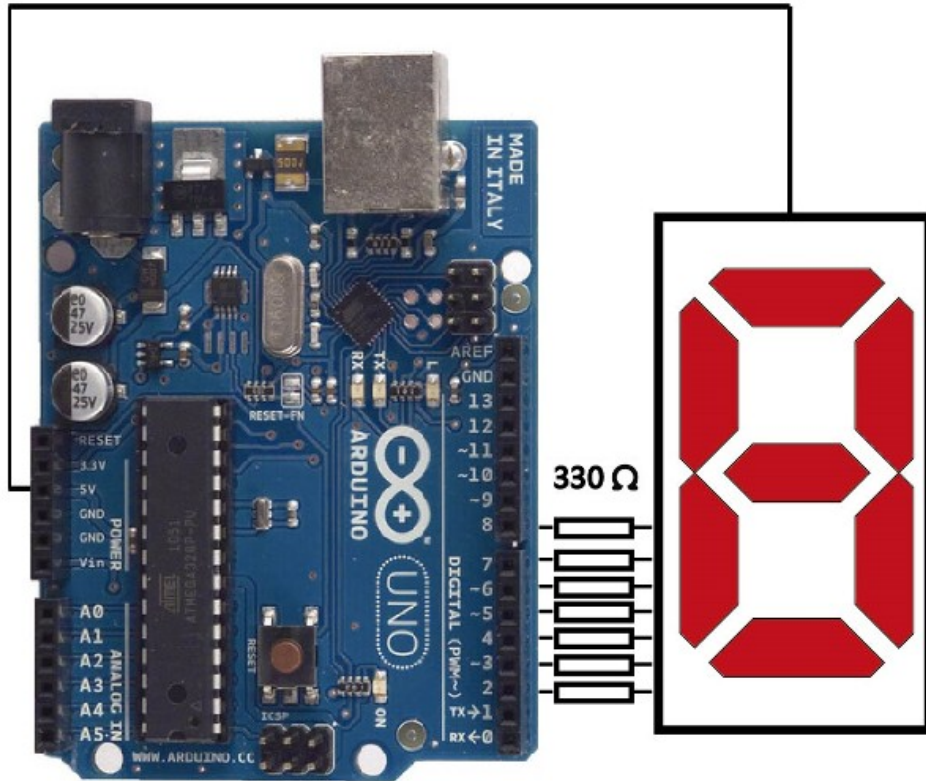


Displays de LED de 7 segmentos

0	a, b, c, d, e, f
1	b, c
2	a, b, d, e, g
3	a, b, c, d, g
4	b, c, f, g
5	a, c, d, f, g
6	a, c, d, e, f, g
7	a, b, c
8	a, b, c, d, e, f, g
9	a, b, c, f, g



Displays de LED de 7 segmentos



Para formar os dígitos, 'acendemos' os segmentos enviando sinais HIGH para as portas em que estiverem ligados (por meio de resistores limitadores).

Precisamos de 8 portas! E, se cada uma consumir 20mA, já são 160mA para um display totalmente aceso (o limite para o Arduino são 200mA).

Controle

display7SegmentosDireto

```
1 //Não abuse da corrente!  
2 //Use resistores de 300ohms ou mais entre o Arduino e o display  
3 //O código pressupõe um display de cátodo comum (negativo comum)  
4 #define a 2  
5 #define b 3  
6 #define c 4  
7 #define d 5  
8 #define e 6  
9 #define f 7  
10 #define g 8  
11 #define pd 9  
12  
13 void setup()  
14 {  
15     pinMode(a, OUTPUT);  
16     pinMode(b, OUTPUT);  
17     pinMode(c, OUTPUT);  
18     pinMode(d, OUTPUT);  
19     pinMode(e, OUTPUT);  
20     pinMode(f, OUTPUT);  
21     pinMode(g, OUTPUT);  
22     pinMode(pd, OUTPUT);  
23 }
```

Ligação das portas para conexão
aos segmentos

Cuidado com a corrente!

Controle

```
25 void loop()
26 {
27     //escreve um 'zero' com o ponto decimal desligado
28     digitalWrite(a, 1);
29     digitalWrite(b, 1);
30     digitalWrite(c, 1);
31     digitalWrite(d, 1);
32     digitalWrite(e, 1);
33     digitalWrite(f, 1);
34     digitalWrite(g, 0);
35     digitalWrite(pd, 0);
36     delay(2000);
37     //escreve um 'três' com o ponto decimal ligado
38     digitalWrite(a, 1);
39     digitalWrite(b, 1);
40     digitalWrite(c, 1);
41     digitalWrite(d, 1);
42     digitalWrite(e, 0);
43     digitalWrite(f, 0);
44     digitalWrite(g, 1);
45     digitalWrite(pd, 1);
46     delay(2000);
```

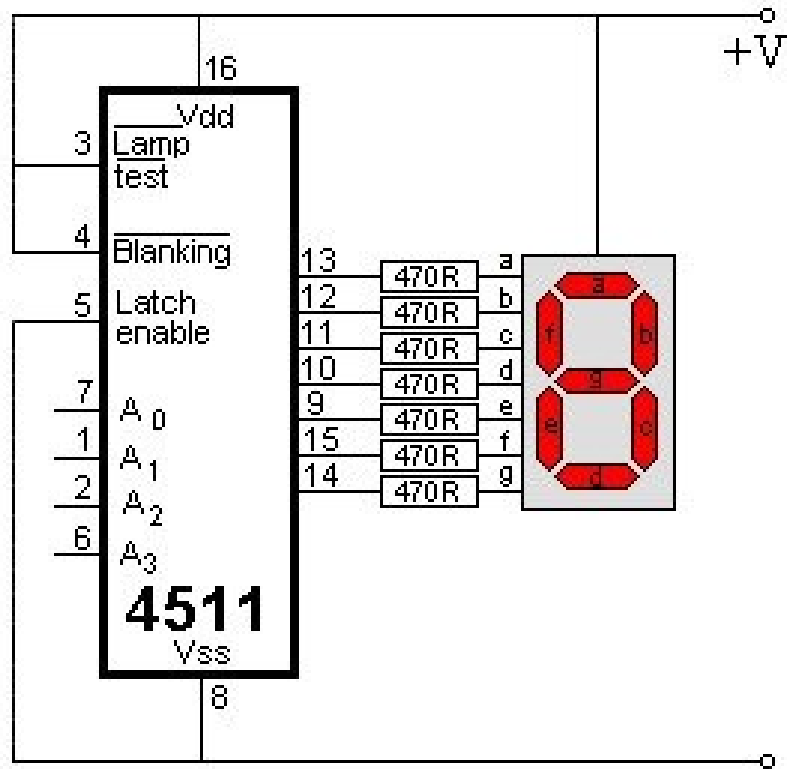
SE for com cátodo comum (negativo comum, GND), ao enviarmos um sinal alto (1) para a porta do segmento, ele acenderá

Controle

SE for com ânodo comum (positivo comum, VCC), ao enviarmos um sinal baixo (0) para a porta do segmento, ele acenderá

```
47  /*SE seu display for de ânodo comum, positivo comum
48     o controle fica invertido. Exemplo para o 'zero':
49     digitalWrite(a, 0);
50     digitalWrite(b, 0);
51     digitalWrite(c, 0);
52     digitalWrite(d, 0);
53     digitalWrite(e, 0);
54     digitalWrite(f, 0);
55     digitalWrite(g, 1);
56     digitalWrite(pd, 1);
57  */
```


Exemplo de controle com CD4511

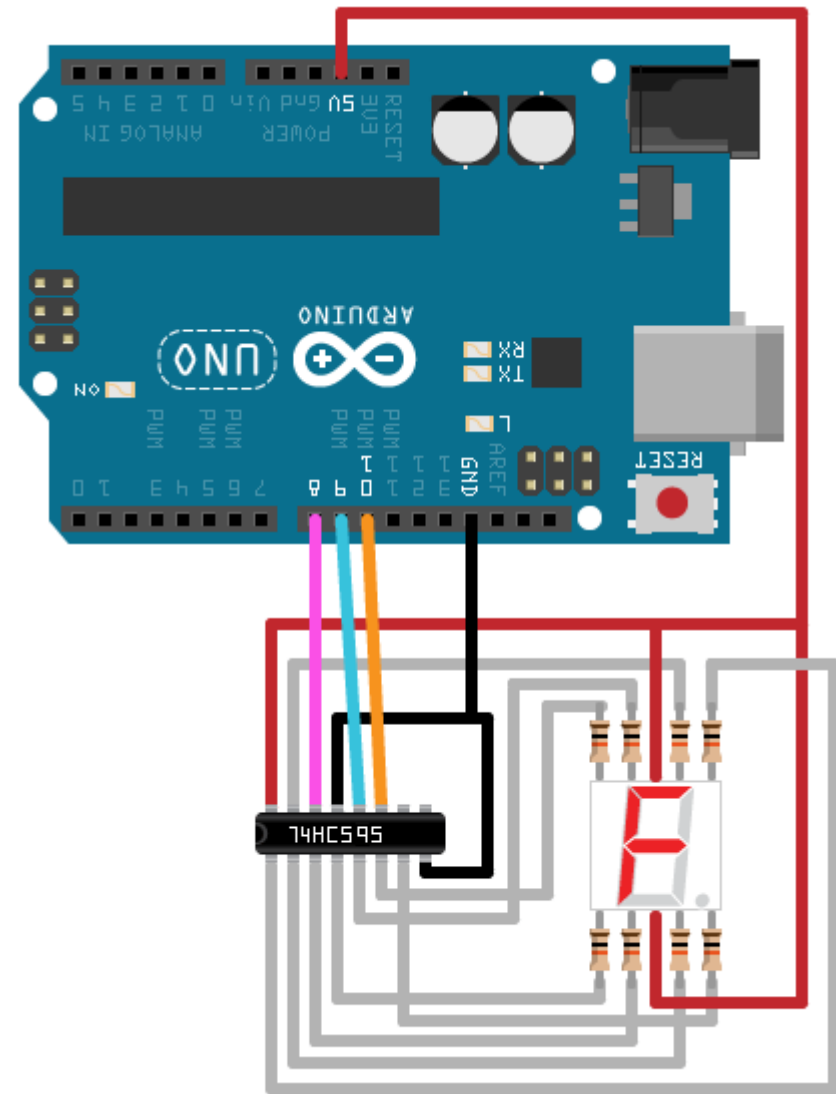


Utiliza 4 sinais (A0 até A3) para controle dos segmentos:

Decimal	A ₀	A ₃	A ₂	A ₁
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

A energia que alimenta o *display* pode vir de uma fonte separada do Arduino.

Controle por meio de circuito integrado
74HC595 – 3 fios de controle; expansão mais
fácil



74HC595

display7SegmentosHC595

```
1 //Controle do display
2 #define latch 3 // pino 12 do 74HC595
3 #define data 4 // pino 14 do 74HC595
4 #define clk 2 // pino 11 do 74HC595
5
6 // dígitos no display (0 a F)
7 int digitos[] = {252, 96, 218, 242, 102, 182, 190, 224, 254, 246, 238, 62, 156, 122, 158, 142};
8
9 void setup(){
10     //controle do display
11     pinMode(data, OUTPUT);
12     pinMode(latch, OUTPUT);
13     pinMode(clk, OUTPUT);
14 }
15
16 void loop(){
17     //escreve zero no display
18     digitalWrite(latch, LOW);
19     shiftOut(data, clk, LSBFIRST, digitos[0]);
20     digitalWrite(latch, HIGH);
21     delay(1000);
22     //escreve oito no display
23     digitalWrite(latch, LOW);
24     shiftOut(data, clk, LSBFIRST, digitos[8]);
25     digitalWrite(latch, HIGH);
26     delay(1000);
27 }
```

Os dígitos são formados no display informando-se quais segmentos estarão acesos e quais não.

O vetor digitos contém todos e é selecionado a partir do índice '0'. Contém 16 dígitos, portanto poderá mostrar códigos hexadecimais entre '0' e 'F' no display.