

Componentes Eletronicos

Guia Completo do Display LCD – Arduino

Eletrogate 13 de abril de 2018

Atualizado em: 14 ago 2023

13 min

Introdução sobre Displays LCD

Os Displays LCD são muito úteis para quem pretende usar um [Micro-controlador](#) para desenvolver uma aplicação. Eles permitem uma interface visual entre homem e máquina (HMI em inglês), barata e simples de usar. No LCD você pode enviar textos, números, símbolos e até imagens que podem dar uma indicação do que o Micro-controlador está fazendo, dos dados que podem estar sendo coletados ou transmitidos, etc. Para quem não sabe, LCD significa em inglês – *Liquid Crystal Display* – ou mostrador de cristal líquido. Uma grande vantagem dos LCDs é que não precisam de muita energia para funcionar. Lembram-se dos relógios digitais de LCD, há décadas atrás? As pilhas podiam durar meses e até um ano. Mas os LCDs que estamos falando, tem LEDs atrás para que possam ser usados em ambientes com pouca ou nenhuma luz. Esses LEDs gastam um pouco mais. Mas se fosse usar somente os LCDs, o consumo de energia seria baixíssimo!

Os monitores dos primeiros Laptops da década de 1980 usavam a mesma tecnologia LCD! E sem iluminação de LED.



Laptop Toshiba T1000 – foto Wikipedia

No início da era das TVs Digitais, essa tecnologia LCD também foi muito usada. Atualmente existem outras tecnologias que reproduzem as imagens com muito mais fidelidade.

O mercado de displays LCD está sempre inovando, e hoje já temos displays coloridos: azuis, vermelhos, verdes, etc, isto é, para todos os gostos. Esse Guia fará sempre referência para os LCDs azuis. Mas fique sabendo que todos tem o mesmo princípio de funcionamento. O que muda somente é a cor do LED que o ilumina.

Para o uso de Micro-controladores como o [Arduino](#), os displays LCD mais comuns são o 16×2 (16 caracteres x 2 linhas) ou 20×4 (20 caracteres x 4 linhas).

[LCD 16x2 azul](#)

[LCD 16x2 verde](#)

[LCD 20x4 azul com I2C](#)

[LCD 20x4 verde](#)

Como Funciona o Display LCD

Quando eles foram criados décadas atrás, eles não eram luminosos. Eles simplesmente deixavam passar a luz ou então a obstruíam, deixando sombras. O mostrador é formado de duas placas acrílicas transparentes. Entre essas placas está o cristal líquido. Esse cristal líquido altera o seu comportamento cristalino, dependendo da tensão aplicada entre ele. Os displays, como dá para ver, são formados de vários pontinhos. Cada pontinho pode ficar claro ou escuro, dependendo da polarização da eletricidade de cada um. Sob as placas transparentes, existem uma matriz invisível de conexões que controlam todos esses pontinhos. Quem faz isso, são os chips controladores que ficam por trás do display.

O chip controlador de LCD mais usado atualmente no mundo inteiro, é um que foi desenvolvido pela Hitachi – o HD44780, há muito tempo atrás. Ele é tão bom, que se tornou um padrão. Se quiser aprofundar os seus estudos sobre LCDs, aprendendo à usar todos os inúmeros recursos disponíveis, recomendo a leitura do datasheet.

[Datasheet HD44780](#)

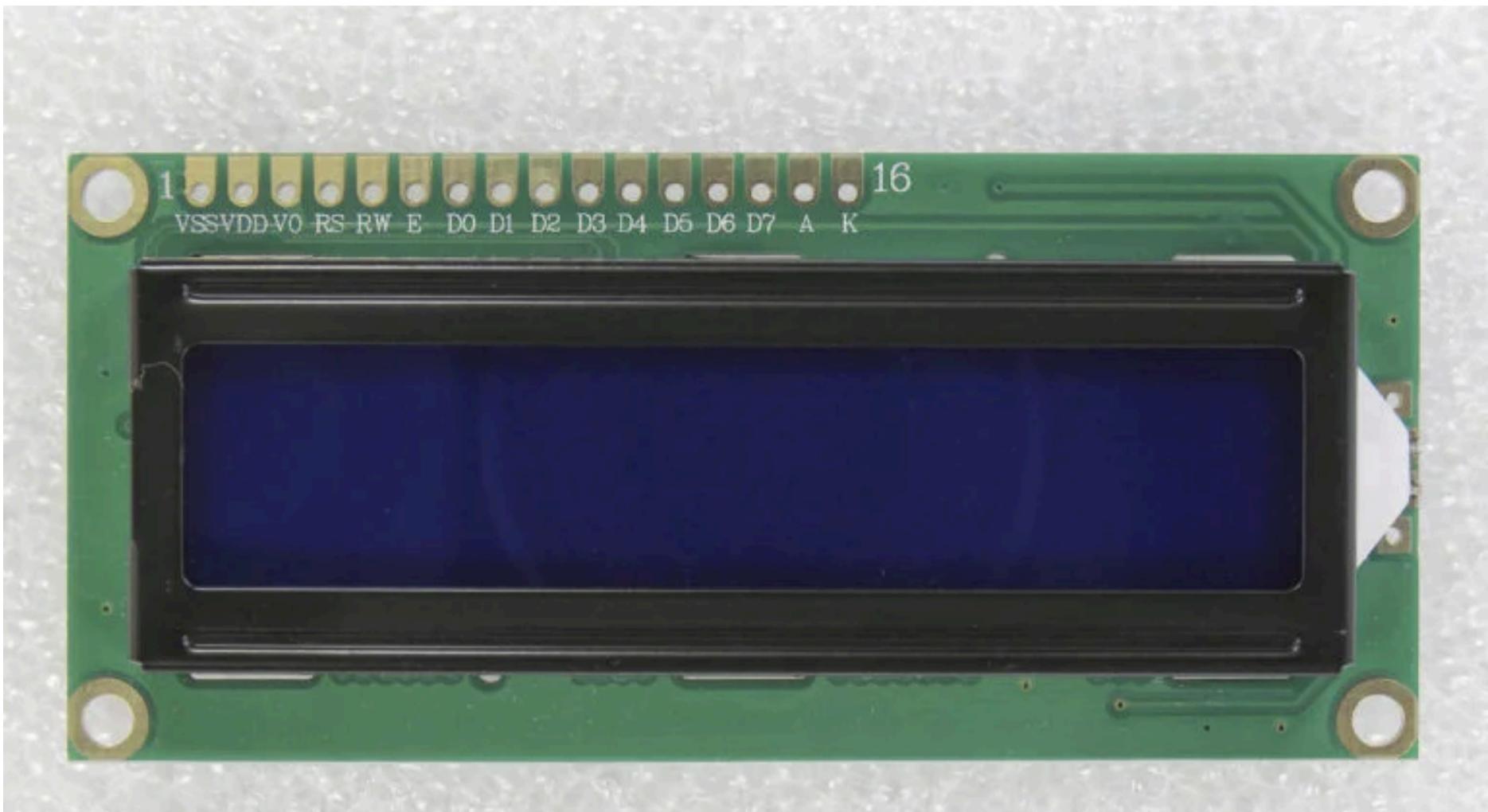
A comunicação entre o Controlador de LCD e o Micro-controlador (por exemplo, Arduino), pode ser paralela (4 ou 8 bits) ou serial (I2C). Para um número limitado de portas digitais, é recomendável o uso da interface I2C.

Como curiosidade, saiba que esse chip possui uma ROM interna onde já estão gravados alguns caracteres e símbolos. E para que você possa gerar seus próprios símbolos, existe uma memória RAM interna. Muito versátil !

Veja uma parte da Tabela de caracteres gravada internamente – cada caracter é formado por uma matriz de 5 x 8 pontos:

Upper 4 bits Lower 4 bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	CG RAM (1)															
xxxx0000	(1)															
xxxx0001	(2)															
xxxx0010	(3)															
xxxx0011	(4)															
xxxx0100	(5)															
xxxx0101	(6)															
xxxx0110	(7)															
xxxx0111	(8)															
xxxx1000	(1)															
xxxx1001	(2)															
xxxx1010	(3)															
xxxx1011	(4)															
xxxx1100	(5)															
xxxx1101	(6)															
xxxx1110	(7)															
xxxx1111	(8)															

Pinagem do LCD



Display LCD 16×2 azul – foto Gustavo Murta

Esse é o Datasheet bem detalhado de um Display LCD 16×2 . Veja que ele usa chips controladores compatíveis com o padrão HD44780 :

[Datasheet LCD 16x2](#)

Na parte superior do [Display](#), existem 16 furos onde podem ser soldados pinos ou cabos de comunicação e alimentação. **Atenção : a ligação incorreta ou uso de tensões acima de 5V poderá danificar o display.**

OBS: A pinagem do LCD 16×2 é idêntica ao LCD 20×4 !

Pinagem e função de cada pino :

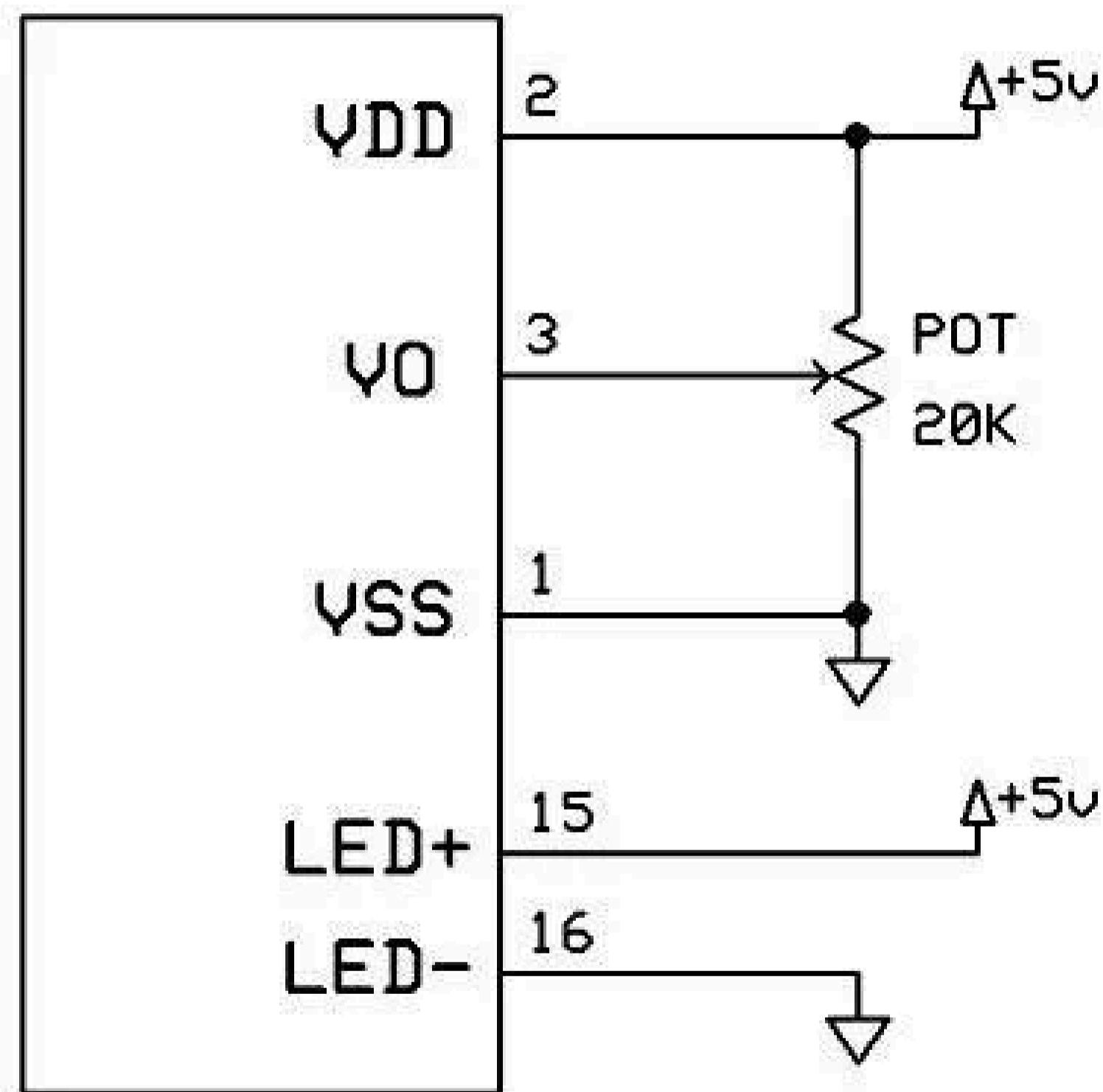
- pino 1 – VSS – Pino de alimentação (zero volts – GND)
- pino 2 – VDD – Pino de alimentação de +5V
- pino 3 – VO – Pino de ajuste do contraste do LCD – depende da tensão aplicada (ajustável)
- pino 4 – RS – Seleção de Comandos (nível 0) ou Dados (nível 1)
- pino 5 – R/W – Read(leitura – nível 1) / Write (escrita – nível 0)
- pino 6 – E – Enable (Ativa o display com nível 1 ou Desativa com nível 0)
- pino 7 – D0 – data bit 0 (usado na interface de 8 bits)
- pino 8 – D1 – data bit 1 (usado na interface de 8 bits)
- pino 9 – D2 – data bit 2 (usado na interface de 8 bits)
- pino 10 – D3 – data bit 3 (usado na interface de 8 bits)
- pino 11 – D4 – data bit 4 (usado na interface de 4 e 8 bits)
- pino 12 – D5 – data bit 5 (usado na interface de 4 e 8 bits)
- pino 13 – D6 – data bit 6 (usado na interface de 4 e 8 bits)
- pino 14 – D7 – data bit 7 (usado na interface de 4 e 8 bits)
- pino 15 – A – Anodo do LED de iluminação (+5V CC)
- pino 16 – K – Catodo do LED de iluminação (GND)

A iluminação do LCD é feita pelo LED. O pino 15 (Anodo do LED) pode ser conectado diretamente em +5 V e o pino 16 (catodo do LED) deve ser conectado no terra (GND). Dessa forma o LED vai consumir aproximadamente 22 mA. O consumo total de corrente (LCD + LED) do LCD 16×2 Azul é de aproximadamente 23 mA (sem considerar o backlight, que pode levar o consumo para entre 40 e 120 mA).

Se deseja consumir menor corrente, use um resistor em série, por exemplo de 220 ohms (consumo 7,6 mA).

Para ajuste do contraste do LCD, a tensão no pino 3 (VO) deve ser ajustada. Use um potenciômetro de 20K ohms. Nas extremidades do POT conecte o +5V e o GND. O pino central conecte no pino 3 do LCD. No meu LCD 16×2 azul, a tensão VO foi igual a 1,0 Volts aproximadamente.

DISPLAY LCD 16x2

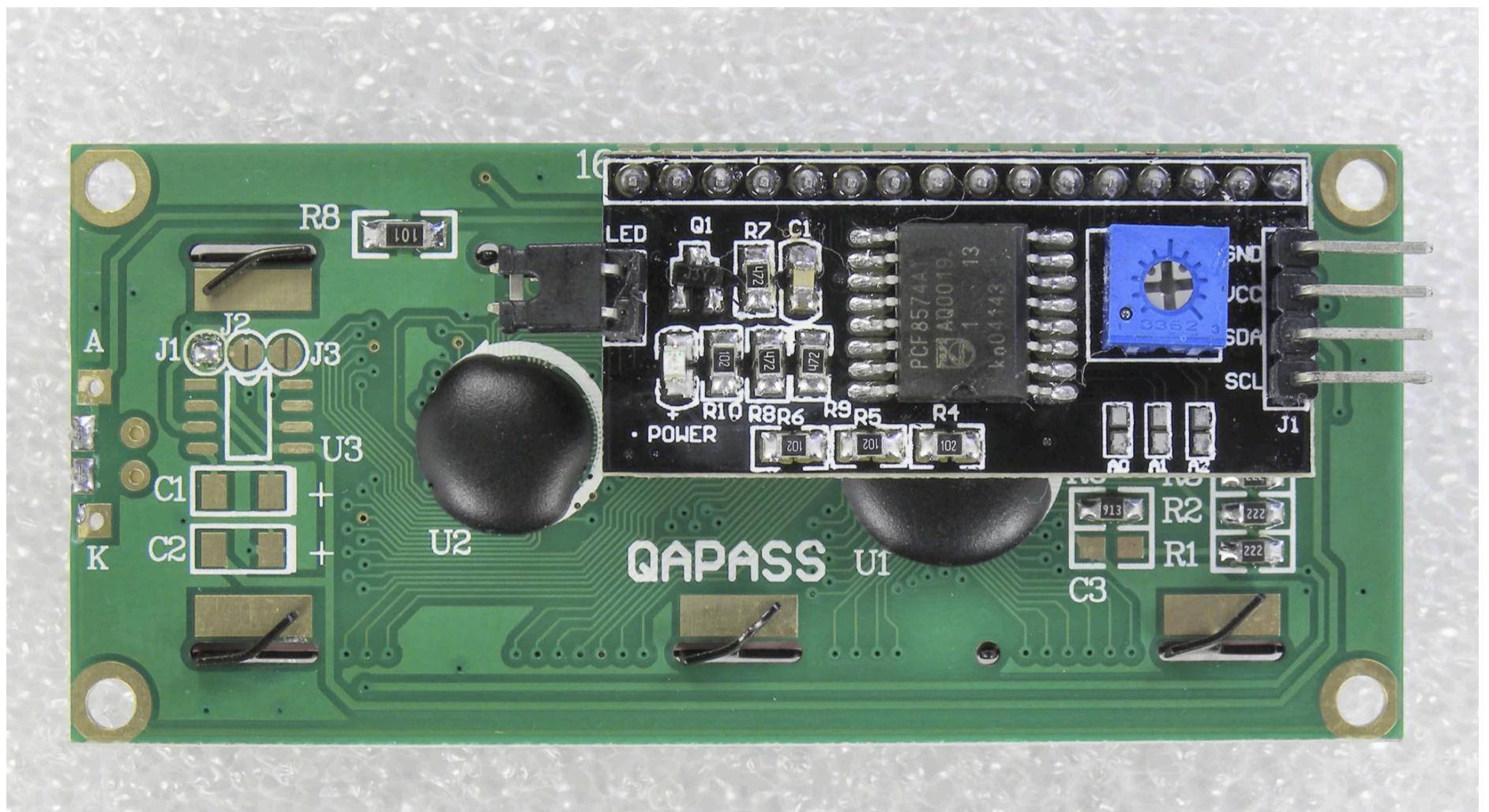


Interface I2C para Displays LCD

Quando esse tipo de Display LCD foi desenvolvido, os barramentos de dados dos Micro-controladores usavam 8 bits. Na era Arduino, as portas digitais disponíveis são reduzidas. Para resolver essa limitação, foi desenvolvida uma interface I2C especialmente dedicada para os LCDs. O chip usado nesse módulo é o [PCF8574](#). Ele é um expansor de

portas paralelas, tem uma interface I2C e pode controlar até 8 bits tanto como entrada ou como saída (dependendo da configuração). A velocidade da interface I2C está limitada a 100 KHz. A tensão de alimentação pode ser 3,3V ou 5V, o que o habilita para todos os Micro-controladores mais comuns.

Essa foto é a parte de trás do Display LCD com a Interface I2C já conectada.



LCD 16×2 com Interface I2C – foto Gustavo Murta

Para conectar com o Arduino ou outro Micro-controlador , somente quatro pinos são necessários :

GND – conecte no terra do Arduino

VCC – conecte na alimentação de 5V

SDA – serial Data – interface I2C

SCL – serial Clock – interface I2C

O **potenciômetro Azul** nessa interface é usado para ajuste do contraste. Após o Display energizado e programado, ajuste-o para tornar a imagem visível. O **jumper LED** é usado para ativar o LED Backlight. Se não quiser usar o LED para economizar energia, retire esse jumper. O **led vermelho** na placa serve como indicação que ela está energizada.

Endereçamento do PCF8574 :

Em uma interface I2C , a comunicação é serial. Para selecionar um dispositivo, o endereço é enviado para o barramento, pois em um mesmo barramento pode-se ter vários dispositivos.

O chip PCF8574 possui alguns endereços já definidos, dependendo do modelo (identifique o chip do seu módulo):

PCF8574 = 0x20H

PCF8574A = 0x38H

Além do endereço básico, através de 3 pinos de endereçamento (A0, A1 e A2) o endereçamento pode ser alterado, como no caso do uso de mais de um chip no mesmo barramento . Veja no módulo, os jumpers A0, A1 e A2.

PCF8574 and PCF8574A I²C-Bus Slave Address Map

INPUTS			PCF8574 I ² C-Bus Slave Address	PCF8574A I ² C-Bus Slave Address
A2	A1	A0		
L	L	L	20 (hexadecimal)	38 (hexadecimal)
L	L	H	21 (hexadecimal)	39 (hexadecimal)
L	H	L	22 (hexadecimal)	3A (hexadecimal)
L	H	H	23 (hexadecimal)	3B (hexadecimal)
H	L	L	24 (hexadecimal)	3C (hexadecimal)
H	L	H	25 (hexadecimal)	3D (hexadecimal)
H	H	L	26 (hexadecimal)	3E (hexadecimal)
H	H	H	27 (hexadecimal)	3F (hexadecimal)

Mais um detalhe muito importante ! Para que a interface I²C funcione adequadamente, resistores de PULL UP devem ser usados nas duas linhas (SCL e SDA) . Esses resistores conectam essas linhas ao VCC . Nesse módulo I²C, esses resistores de PULL UP (4,7 K ohms) já existem. Portanto não acrescente mais resistores na interface.

Display LCD com Arduino (barramento 4 bits)

Como já informei , o Display LCD pode ser conectado ao Arduino através de um barramento **Paralelo** ou **Serial** (usando o módulo I²C).

No caso do barramento paralelo para o Arduino, usa-se apenas 4 bits de dados. Mas para as linhas de controles, mais portas são necessárias (RS, R/W e ENABLE) . Somando um total de 7 portas ! Esse tipo de comunicação deve ser evitado se não houver um número de portas disponíveis para a aplicação.

Materiais necessários para o projeto com Arduino Nano e Display LCD 16x2

[Nano V3 + Cabo Usb para Arduino](#)

[Protoboard 830 Pontos](#)

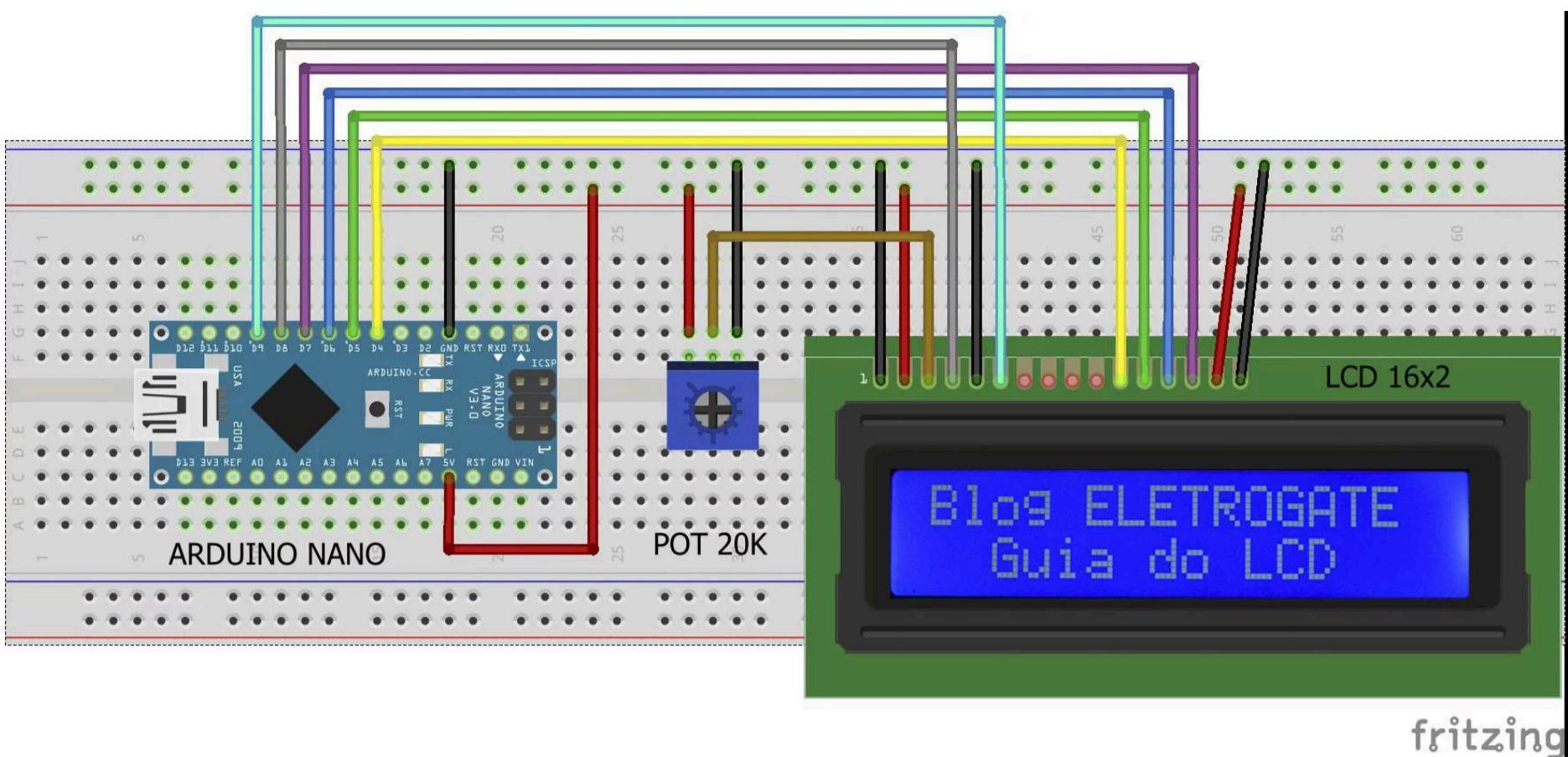
[Jumpers – Macho/Macho – 40 Unidades de 10cm](#)

[Display LCD 16x2 com Backlight Azul](#)

[Potenciômetro Trimpot 10K Horizontal](#)



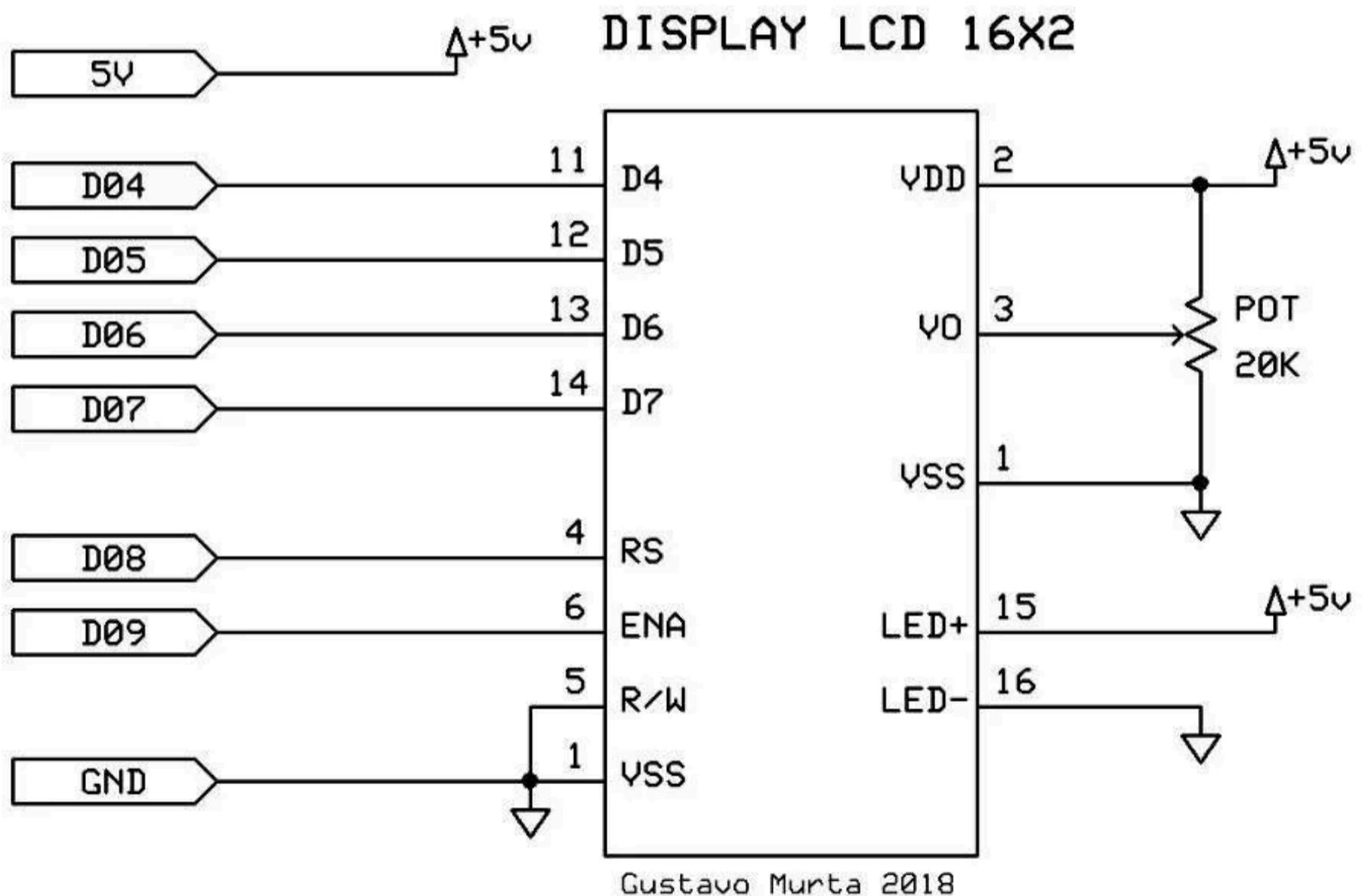
Esse é o diagrama **Fritzing** do circuito para teste do **Display LCD 16x2 com Arduino Nano** :



fritzing

E esse é o diagrama [eletrônico](#) do mesmo circuito – **Arduino Nano com LCD 16×2**:

Veja que o pino 5 (R/W) esta aterrado (nível zero) . Portanto o Display só pode receber dados (Write).



Para posicionar os caracteres no Display, existe esse endereçamento das linhas e das colunas. Veja que a contagem começa do 0 e não do 1 (Linha 0, coluna 0) ! As diferenças de um Display de 16×2 para um Display 20×4 são a quantidade de caracteres e o endereçamento. Todos os comandos são idênticos para os dois tipos de Displays.

Enderecamento de Linhas e Colunas

	Colunas		LCD 16X2														
Linha 0	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
Linha 1	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
Colunas																	

Enderecamento de Linhas e Colunas

	Colunas		LCD 20X4																	
Linha 0	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
Linha 1	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
Linha 2	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
Linha 3	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
Colunas																				

Nessa montagem , usaremos a [Biblioteca Liquid Crystal](#), portanto ela deve ser instalada usando o procedimento abaixo.

Instalando uma nova Biblioteca na Arduino IDE

Para instalar a nova Biblioteca, clique em **Sketch > Incluir Biblioteca > Gerenciar Bibliotecas** .

Após abrir a janela do **Gerenciador de Biblioteca**, refine a busca digitando o nome da biblioteca. Na biblioteca selecionada, clique em **More Info** e depois em **Instalar**. Após alguns segundos, ela será automaticamente instalada. Lembre-se que o seu computador precisa estar conectado na internet. Após a instalação da biblioteca, é necessário que feche e abra novamente o programa **Arduino IDE**.

Usando exemplos da **Biblioteca Liquid Crystal**, fiz esse Sketch para testar várias funções do Display :

- imprimir mensagem
- piscar a mensagem
- teste do cursor
- teste de rolagem da mensagem
- teste de direção da mensagem

Para configurar o tipo de Display :

```
Lcd.begin(16, 2); // definindo o LCD – 16 colunas e 2 linhas
```

```
Lcd.begin(20, 4); // definindo o LCD – 20 colunas e 4 linhas
```

```

1. /* Programa de Teste de LCD
2. Blog Eletrogate - https://blog.eletrogate.com/guia-completo-do-display-lcd-arduino/
3. Arduino Nano - LCD 16/2 azul - IDE 1.8.5
4. Gustavo Murta 11/abril/2018
5. Biblioteca LiquidCrystal https://github.com/arduino-libraries/LiquidCrystal
6. */
7.
8. #include <LiquidCrystal.h>      // usando a biblioteca LiquidCrystal
9.
10. const int rs = 8, en = 9, d4 = 4, d5 = 5, d6 = 6, d7 = 7; // definicao dos pinos do Display
11. LiquidCrystal lcd(rs, en, d4, d5, d6, d7);           // configurando os pinos
12. int thisChar = 0 ;
13.
14. void setup()
15. {
16.     lcd.begin(16, 2);          // definindo o LCD - 16 colunas e 2 linhas
17. }
18.
19. void Hello ()                // imprimindo mensagem
20. {
21.     lcd.setCursor(0, 0);       // selecionando coluna 0 e linha 0
22.     lcd.print("Blog ELETROGATE"); // print da mensagem
23.     lcd.setCursor(2, 1);       // selecionando coluna 2 e linha 1
24.     lcd.print("Guia do LCD"); // Print da mensagem
25.     delay(1000);             // atraso de 1 segundo
26. }
27.
28. void Flash ()
29. {
30.     lcd.noDisplay();          // desliga display
31.     delay(1000);              // atraso de meio segundo
32.     lcd.display();            // liga display
33.     delay(1000);              // atraso de meio segundo
34.     lcd.clear();               // limpa a tela
35.     delay(1000);              // atraso de 1 segundo
36. }
37.
38. void Blink ()                // teste do cursor
39. {
40.     lcd.noBlink();             // apaga cursor
41.     delay(1000);              // atraso de 1 segundo
42.     lcd.blink();               // acende cursor
43.     delay(1000);              // atraso de 1 segundo
44.     lcd.clear();               // limpa a tela
45.     delay(1000);              // atraso de 1 segundo
46. }
47.
48. void AutoScroll ()           // teste de rolagem de mensagem
49. {
50.     lcd.setCursor(16, 1);       // selecionando coluna 16 e linha 1
51.     lcd.autoscroll();          // configura rolagem automatica de mensagem
52.     for (thisChar = 0; thisChar < 10; thisChar++) // imprime de 0 a 9
53.     {
54.         lcd.print(thisChar);    // imprime o numero
55.         delay(350);             // atraso de 350 ms
56.     }
57.     lcd.noAutoscroll();        // desliga rolagem automatica
58.     lcd.clear();               // limpa a tela
59.     delay(1000);              // atraso de 1 segundo
60. }
61.
62. void dirText ()              // teste de direcao de mensagem
63. {
64.     lcd.clear();               // limpa a tela
65.     lcd.cursor();              // liga o cursor
66.     lcd.setCursor(10, 0);       // selecionando coluna 10 e linha 1
67.     for (thisChar = 1; thisChar < 10; thisChar++) // imprime de 1 a 9
68.     {
69.         lcd.rightToLeft();      // imprime da direita para a esquerda
70.         lcd.print(thisChar);    // imprime o numero
71.         delay(350);             // atraso de 350 ms
72.     }
73.     for (thisChar = 1; thisChar < 10; thisChar++) // imprime de 1 a 9
74.     {
75.         lcd.leftToRight();      // imprime da esquerda para a direita
76.         lcd.print(thisChar);    // imprime o numero
77.         delay(350);             // atraso de 350 ms
78.     }
79.     lcd.noCursor();            // desliga o cursor
80. }
81.
82. void loop()

```

```

83. {
84.     Hello ();           // imprimindo mensagem
85.     Flash ();          // piscando a mensagem
86.     Blink ();          // teste do cursor
87.     AutoScroll ();     // teste de rolagem de mensagem
88.     dirText ();        // teste de direcao de mensagem
89.     delay(1000);       // atraso de 1 segundo
90. }
```

Criando Caracteres Especiais para o Display LCD

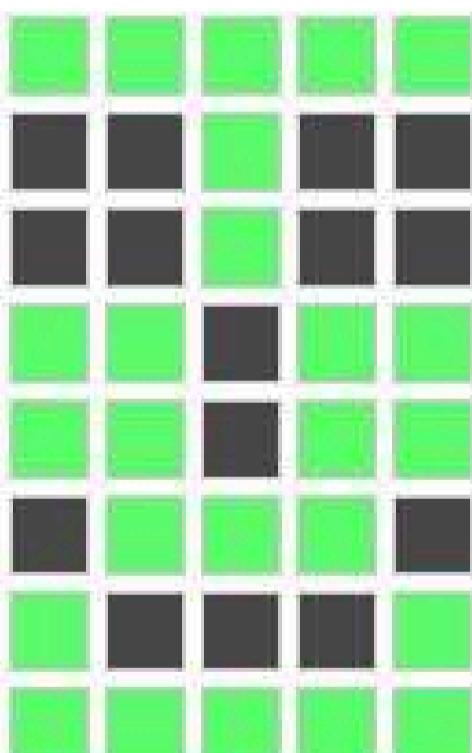
Mais um recurso bem legal da [Biblioteca Liquid Crystal](#): criar caracteres especiais ! Essa biblioteca tem um método que permite, através do controle dos pontos da matriz 5×8 de cada caracter do LCD, a criação de qualquer símbolo ou caracter especial. Use o Gerador de caracteres do link abaixo, para criar o código do Sketch. Para desenhar, clique nos pontos da matriz 5×8 . Copie o código gerado e insira-o no Sketch.

[Gerador de símbolos e caracteres especiais](#)

Custom Character Generator for HD44780 LCD Modules

Click pixels to generate output.

Pixels



Clear

Invert

Output

```

byte customChar[8] = {
    0b000000,
    0b11011,
    0b11011,
    0b00100,
    0b00100,
    0b10001,
    0b01110,
    0b000000
};
```

[LCD16x2ArduinoCaracter.ino](#)

```
1. /* Programa Gerador de Caracter Especial no LCD
```

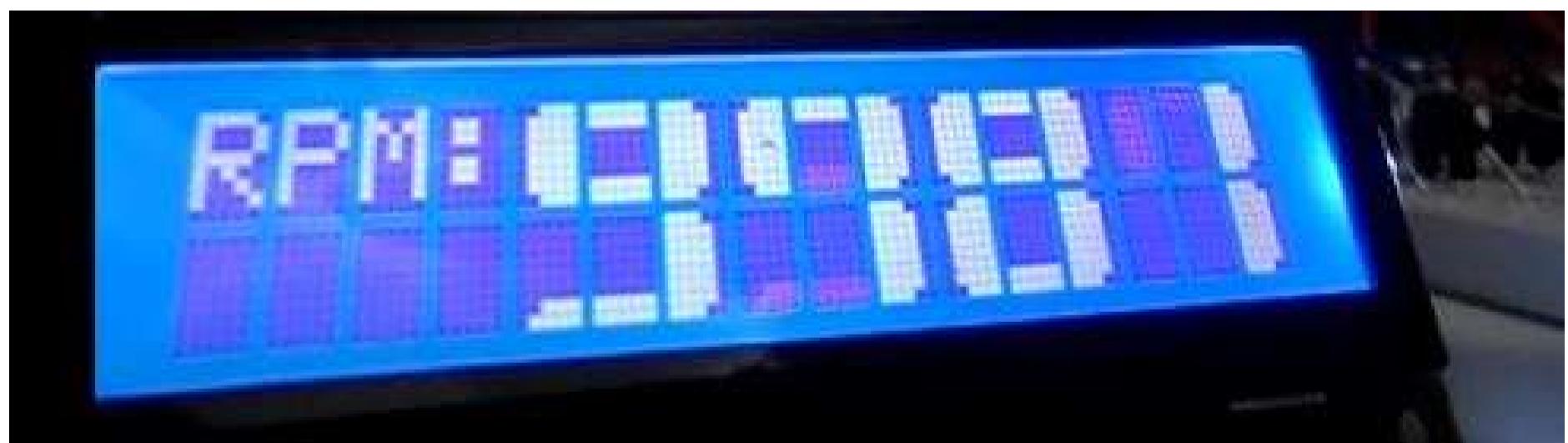
```

2.     Blog Eletrogate - https://blog.eletrogate.com/guia-completo-do-display-lcd-arduino/
3.     Arduino Nano - LCD 16/2 azul - IDE 1.8.5
4.     Gustavo Murta    12/abril/2018
5.     Biblioteca LiquidCrystal https://github.com/arduino-libraries/LiquidCrystal
6. */
7.
8. #include <LiquidCrystal.h>           // usando a biblioteca LiquidCrystal
9.
10. const int rs = 8, en = 9, d4 = 4, d5 = 5, d6 = 6, d7 = 7; // definicao dos pinos do Display
11. LiquidCrystal lcd(rs, en, d4, d5, d6, d7);           // configurando os pinos
12.
13.
14. byte customChar[8] =                  // Matriz do caracter especial
15. {
16.     0b00000,
17.     0b11011,
18.     0b11011,
19.     0b00100,
20.     0b00100,
21.     0b10001,
22.     0b01110,
23.     0b00000
24. };
25.
26. void setup()                         // ate 8 caracteres especiais podem ser criados
27. {
28.     lcd.clear();                      // limpa a tela
29.     lcd.createChar(1, customChar);    // criando o caracter especial 1
30.     lcd.begin(16, 2);               // definindo o LCD - 16 colunas e 2 linhas
31.     lcd.write((byte)1);              // imprimindo o carcter especial 1
32. }
33.
34. void loop()
35. {
36. }
```

Encontrei no Youtube, essa aplicação também bem interessante do uso de números grandes no Display LCD 16x2.

[RPM Large Font \(Ronivaldo\)](#)

Biblioteca Big Crystal



A biblioteca Liquid Crystal é bem versátil ! Permite o uso de vários tipos de interface :

Interface paralela 4 bits

Interface Serial I2C

Interface SPI

Como referência para consulta, esse é o link para uso da interface SPI (pouco usada):

[LCD16x2 com Interface SPI - ARDUINO](#)

Display LCD com Arduino (Interface I2C)

Como já citei anteriormente, existe um módulo de Interface I2C especialmente projetado para o uso de Displays LCD. A grande vantagem é o número reduzido de portas usadas no Arduino – somente duas – SCL e SDA.

[Arduino – Biblioteca Wire / I2C](#)

Ligações das Portas I2C (para Arduino) :

Porta SCL = pino A5

Porta SDA = pino A4

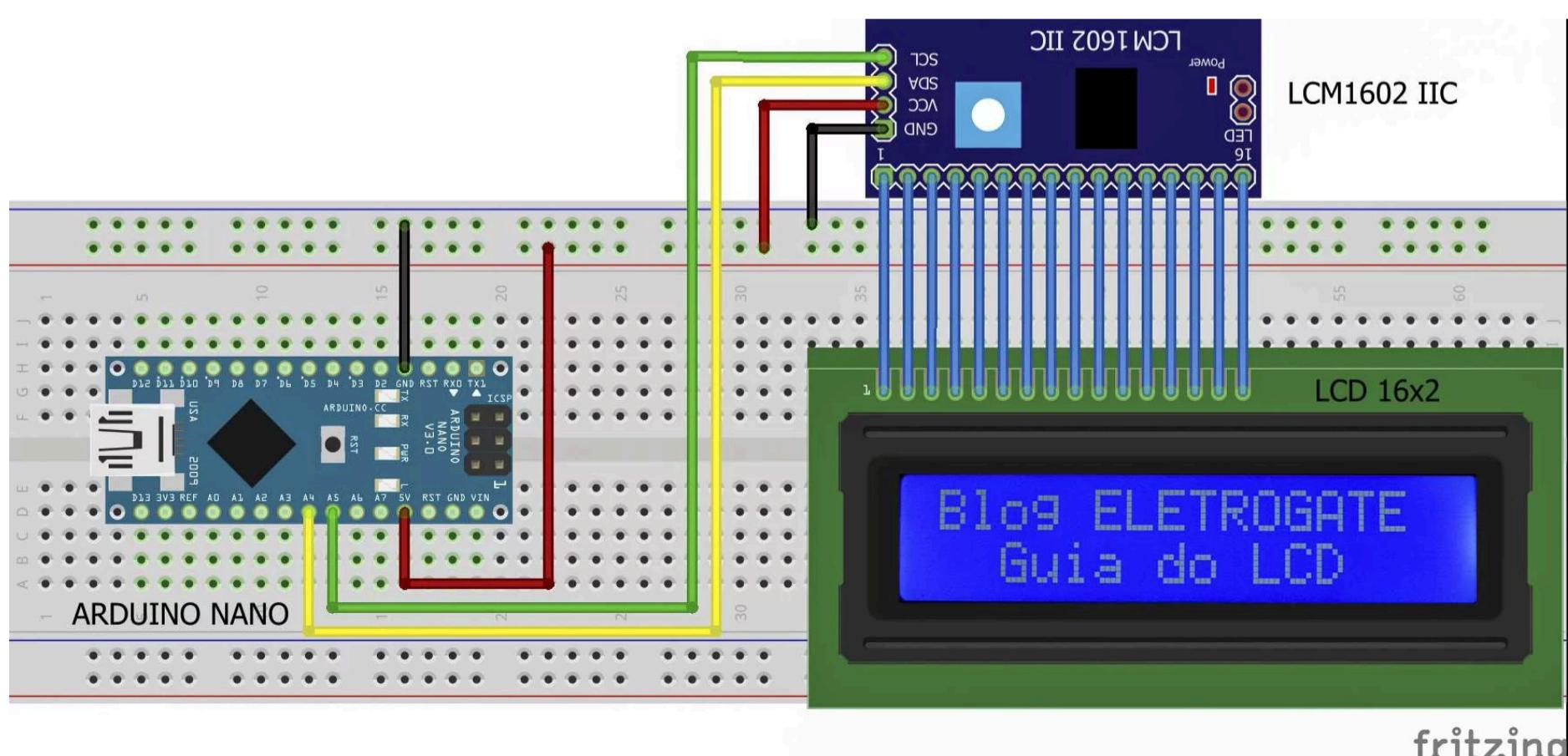
Não se esqueça de conectar o **GND** da Interface I2C no **GND** do Arduino. O mesmo para **5V**. O consumo de corrente medido por mim foi de **27,5 mA**. Faça o **ajuste do contraste** usando o potenciômetro da Interface I2C.

Materiais necessários para o projeto com Arduino Nano e Display LCD 16x2 I2C

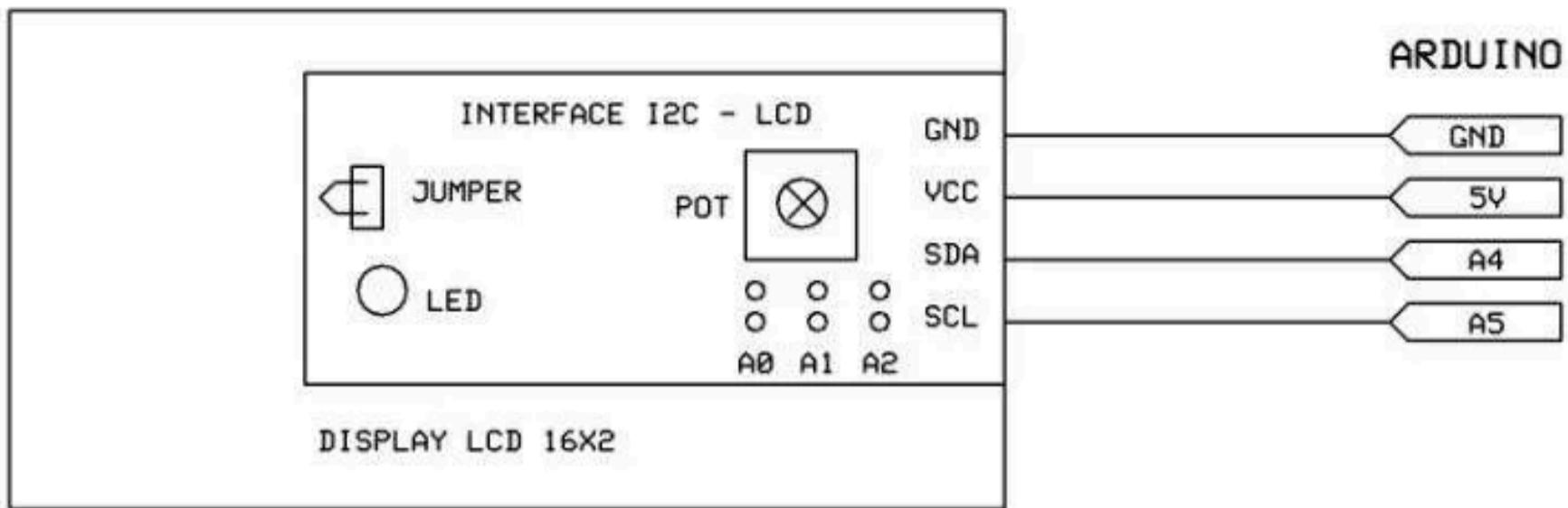
- 1x [Nano V3.0 + Cabo Usb para Arduino](#)
- 1x [Protoboard 830 Pontos](#)
- 1x [Jumpers – Macho/Femea – 20 Unidades de 20cm](#)
- 1x [Display LCD 16x2 com Backlight Azul](#)
- 1x [Módulo Serial I2C para Display LCD](#)



Esse é o diagrama **Fritzing** do circuito para teste do **Display LCD 16x2 (Interface I2C) com Arduino Nano** :



Esse é o diagrama eletrônico do mesmo circuito – **Display LCD 16x2 (Interface I2C) com Arduino Nano**:



I2C Scanner

O primeiro teste a ser feito, é a identificação do endereço I2C da interface. Rode o Sketch **I2C Scanner**. Na minha montagem, o endereço encontrado foi:

I2C scanner. Procurando ...

Endereço I2C encontrado: 63 (0x3F) – esse endereço deverá ser inserido no Sketch de teste

Encontrado 1 dispositivo(s).

[I2CscannerEC.ino](#)

```
1. // I2C Scanner
2. // Written by Nick Gammon
3. // Date: 20th April 2011
4.
5. #include <Wire.h>
6.
7. void setup()
8. {
9.   Serial.begin (9600);
10.  Serial.println ();
11.  Serial.println ("I2C scanner. Procurando ...");
12.  byte count = 0;
13.
14. Wire.begin();
15. for (byte i = 8; i < 120; i++)
16. {
17.   Wire.beginTransmission (i);
18.   if (Wire.endTransmission () == 0)
19.   {
20.     Serial.print ("Endereço I2C encontrado: ");
21.     Serial.print (i, DEC);
22.     Serial.print (" (0x");
23.     Serial.print (i, HEX);
24.     Serial.println ());
25.     count++;
26.     delay (1);
27.   }
28. }
29. Serial.print ("Encontrado ");
30. Serial.print (count, DEC);
31. Serial.println (" dispositivo(s).");
32. }
33.
34. void loop() {}
```

Essa é a Biblioteca **Liquid Crystal I2C** usada nessa montagem de teste (instale-a usando o procedimento já citado anteriormente):

Biblioteca Liquid Crystal I2C

Usando a Biblioteca **Liquid Crystal I2C**, criei o programa para testar as mesmas funções do exemplo anterior (interface 4 bits) :

```
imprimir mensagem  
piscar a mensagem  
teste do cursor  
teste de rolagem da mensagem  
teste de direção da mensagem
```

LCD16x2ArduinoI2C.ino

```
1.  /* Programa de Teste de LCD - Interface I2C  
2.   Blog Eletrogate - https://blog.eletrogate.com/guia-completo-do-display-lcd-arduino/  
3.   Arduino Nano - LCD 16/2 azul - IDE 1.8.5  
4.   Gustavo Murta 13/abril/2018  
5.   Biblioteca LiquidCrystal I2C https://github.com/marcoschwartz/LiquidCrystal\_I2C  
6. */  
7.  
8.  
9. #include <Wire.h>           // usando a biblioteca Wire  
10. #include <LiquidCrystal_I2C.h> // usando a biblioteca LiquidCrystal I2C  
11.  
12. LiquidCrystal_I2C lcd(0x3F, 16, 2); // Configura endereço I2C e display com 16 caracteres e 2 linhas  
13. int thisChar = 0 ;  
14.  
15. void setup()  
16. {  
17.   lcd.init();           // inicializa LCD  
18.   lcd.backlight();      // ativa led de backlight  
19. }  
20.  
21. void Hello ()          // imprimindo mensagem  
22. {  
23.   lcd.setCursor(0, 0);  // selecionando coluna 0 e linha 0  
24.   lcd.print("Blog ELETROGATE"); // print da mensagem  
25.   lcd.setCursor(2, 1);  // selecionando coluna 2 e linha 1  
26.   lcd.print("Guia do LCD"); // Print da mensagem  
27.   delay(1000);         // atraso de 1 segundo  
28. }  
29.  
30. void Flash ()  
31. {  
32.   lcd.noDisplay();     // desliga display  
33.   delay(1000);         // atraso de meio segundo  
34.   lcd.display();       // liga display  
35.   delay(1000);         // atraso de meio segundo  
36.   lcd.clear();         // limpa a tela  
37.   delay(1000);         // atraso de 1 segundo  
38. }  
39.  
40. void Blink ()          // teste do cursor  
41. {  
42.   lcd.noBlink();        // apaga cursor  
43.   delay(1000);          // atraso de 1 segundo  
44.   lcd.blink();          // acende cursor  
45.   delay(1000);          // atraso de 1 segundo  
46.   lcd.clear();          // limpa a tela  
47.   delay(1000);          // atraso de 1 segundo  
48. }  
49.  
50. void AutoScroll ()     // teste de rolagem de mensagem  
51. {  
52.   lcd.setCursor(16, 1); // selecionando coluna 16 e linha 1  
53.   lcd.autoscroll();    // configura rolagem automatica de mensagem  
54.   for (thisChar = 0; thisChar < 10; thisChar++) // imprime de 0 a 9  
55.   {
```

```

56.     lcd.print(thisChar);           // imprime o numero
57.     delay(350);                 // atraso de 350 ms
58. }
59. lcd.noAutoscroll();            // desliga rolagem autoamtica
60. lcd.clear();                  // limpa a tela
61. delay(1000);                 // atraso de 1 segundo
62. }
63.
64. void dirText ()                // teste de direcao de mensagem
65. {
66.     lcd.clear();                // limpa a tela
67.     lcd.cursor();               // liga o cursor
68.     lcd.setCursor(10, 0);        // selecionando coluna 10 e linha 1
69.     for (thisChar = 1; thisChar < 10; thisChar++)    // imprime de 1 a 9
70.     {
71.         lcd.rightToLeft();      // imprime da direita para a esquerda
72.         lcd.print(thisChar);    // imprime o numero
73.         delay(350);             // atraso de 350 ms
74.     }
75.     for (thisChar = 1; thisChar < 10; thisChar++)    // imprime de 1 a 9
76.     {
77.         lcd.leftToRight();      // imprime da esquerda para a direita
78.         lcd.print(thisChar);    // imprime o numero
79.         delay(350);             // atraso de 350 ms
80.     }
81.     lcd.noCursor();            // desliga o cursor
82. }
83.
84. void loop()
85. {
86.     Hello ();                  // imprimindo mensagem
87.     Flash ();                  // piscando a mensagem
88.     Blink ();                  // teste do cursor
89.     AutoScroll ();             // teste de rolagem de mensagem
90.     dirText ();                // teste de direcao de mensagem
91.     delay(1000);               // atraso de 1 segundo
92. }

```

[Conheça a Metodologia Eletrogate e ofereça aulas de robótica em sua escola!](#)

Sobre o Autor



José Gustavo Abreu Murta

Consultor e Projetista de Sistemas Embarcados. Técnico em eletrônica, formado em Curso superior de TPD, pós-graduado em Marketing. Trabalhou por muitos anos na IBM na área de manutenção de computadores de grande porte. Aposentou-se, podendo curtir o que mais gosta : estudar e ensinar Tecnologia. Hobista em eletrônica desde 1976. Gosta muito de Fotografia e Observação de aves.



Eletrogate

13 de abril de 2018 Atualizado em: 14 ago 2023

G

Participe da discussão...

FAZER LOGIN COM

OU REGISTRE-SE NO DISQUS 

Nome



Compartilhar

[Mais votados](#)[Mais recentes](#)[Mais antigos](#)**Marcio Mello**

3 anos atrás

Parabéns pelo conteúdo!
Ia ser ótimo um conteúdo sobre displays inteligentes, em Gustavo? XD
Abraço
<https://victorvision.com.br...>

1 0 Responder Compartilhar >

**Lauro Becker**

2 meses atrás

Caro Murta, mais uma excelente e didática aula. Sem me aprofundar muito no assunto, estou com dificuldade com acentuação em português no display....

(A eletrogate está de parabéns por fornecer aos usuários esses exemplos práticos dos produtos.
Saudades também do bom Lab de Garagem...)

0 0 Responder Compartilhar >

E

Eduardo Passarelli

um ano atrás

Parabéns pela matéria, bem completa e rica em informações!

0 0 Responder Compartilhar >

J

José Leonardo

um ano atrás

Boa tarde, estou precisando do esquematico do LCD 16x2 junto com o modulo I2C, para o Kicad. Voce sabe onde encontro?

0 0 Responder Compartilhar >

**Egnaldo Menegassi**

3 anos atrás

Sobre a comunicação I2C, vc tem alguma biblioteca que não conflita quando se usa servo motor junto?
talvez seria algo muito importante para mencionar nesse guia que ta bem legal.

0 0 Responder Compartilhar >

**Jose Gustavo Abreu Murta**

→ Egnaldo Menegassi

3 anos atrás

Olá Egnaldo,
De qual biblioteca servo se refere?
Atenciosamente,
Gustavo Murta

0 0 Responder Compartilhar >

A

Augusto Alves

4 anos atrás

Ótimo artigo, parabéns!!

0 0 Responder Compartilhar >

**Leandro**

4 anos atrás

Muito bom este artigo, parabens!!!

0 0 Responder Compartilhar >

 Jose Gustavo Abreu Murta

→ Leandro



JOSÉ GUSTAVO ABREU MURTA

Leandro

4 anos atrás

Bom dia Leandro,
Muito obrigado pelo elogio.
Atenciosamente,
Gustavo Murta

0 0 Responder Compartilhar >



Guilherme Martins Feliciano

4 anos atrás editado

Tenho uma pergunta. Possuo um LCD 2004 com 20 pinos e só acho LCD com 16 pinos. Como não entendo de LCD gostaria de saber se tem como instalar esse de 16 ou realmente tenho que achar um de 20 pinos.

0 0 Responder Compartilhar >



José Gustavo Abreu Murta

Guilherme Martins Feliciano

4 anos atrás

Olá Gilherme,
Envie as especificações do seu display LCD. Se não tiver, envie fotos detalhadas do mesmo.
Veja um modelo de LCD 2004:
<http://wiki.sunfounder.cc/i...>
Atenciosamente,
Gustavo Murta

0 0 Responder Compartilhar >



Nsn do Brasil

5 anos atrás

Tenho uma duvida como ligar para poder fazer alguns teste ou mesmmo saber se estar bom funcionando este display
LC-Display Toshiba TLX-711A-E0 [Visualizar](#) – uploads.disquscdn.com [Visualizar](#) – uploads.disquscdn.com
Eu tenho um Arduino UNO e um Arduino NANO e um programador de PICs.
Pergunto como ligar ele ?
É possivel ligar com módulo I2C ?
Agradeço qualquer ajuda .
Oliva

0 0 Responder Compartilhar >



José Gustavo Abreu Murta

Nsn do Brasil

5 anos atrás editado

Boa noite NSN,

Descobri através do Google, que esse display pode ser usado no Sintetizador Roland D-70 de 1990.

É um display muito antigo.

Achei isso <http://mdfs.net/Info/Comp/L...>

+-----+

| TOSHIBA DOT MATRIX LCD MODULE

|

| ***** TLX-711A *****

|

| >GRAPHICS DISPLAY : 240X64 DOT

| >CHARACTER DISPLAY : 40Chara.X 8 Lines

| *BUILT-IN LCD CONTROLLER T6963C

| *BUILT-IN DISPLAY RAM 8k Byte

+-----+

Verifique se um desses chips é realmente o controlador T6963C. Se for veja o datasheet:

<https://www.sparkfun.com/datasheets/TI/TLX711A.pdf>

Achei isso também :

<https://www.instructables.com/id/How-to-Interface-a-TOSHIBA-TLX-711A-LCD/>

Você precisará estudar o datasheet dele para descobrir como ele funciona. E depois desenvolver um programa dedicado para o controle. Tarefa arrojada! Boa sorte!

Atenciosamente,

Gustavo Murta

0 0 Responder Compartilhar >



Wilson Roberto Paula Souza

5 anos atrás

Olá,

Estou encontrando dificuldades para usar o LCD 16x2, no ESP8266, seria em face da tensão (3.3V)?

Já revisei a biblioteca e sempre aparece Erro.

0 0 Responder Compartilhar >



Jose Gustavo Abreu Murta

→ Wilson Roberto Paula Souza

5 anos atrás

Boa noite,

Eu nunca testei o LCD com o ESP8266. Sugiro que leia esse tutorial:

<https://randomnerdtutorials...>

Atenciosamente,

Gustavo Murta

0 0 Responder Compartilhar >



Silvano Vieira

5 anos atrás

Boa tarde, amigo. Achei a resposta nessa página: <https://randomnerdtutorials...>

O interessante é que eu já tinha lido antes o conteúdo dessa página e não vi..rssss...mesmo assim, obrigado!

0 0 Responder Compartilhar >



Jose Gustavo Abreu Murta

→ Silvano Vieira

5 anos atrás

Boa noite,

Esse tutorial do Rui Santos é excelente. Eu me basei nele para fazer esse outro tutorial:

<https://blog.eletrogate.com...>

Atenciosamente,

Gustavo Murta

0 0 Responder Compartilhar >



Silvano Vieira

→ Jose Gustavo Abreu Murta

5 anos atrás editado

Boa noite! Também gostei, aprendi bastante lá....Estou alterando um programa de Termômetro Eletrônico que funcionou muito bem no display LCD, inclusive usando seu modelo para criar caracteres especiais. O único problema, além do tamanho do OLED, é que, quando a temperatura muda, ele escreve os números por cima do anterior, sem o apagar, ficando "embolado"! Estou queimando meus poucos neurônios para resolver....obrigado pela atenção e pela resposta..abraços e parabéns pelos seus posts!

PS.: Programando em C eu resolvi problema parecido usando um comando existente para apagar a linha toda ou a partir da posição do cursor. Na biblioteca Adafruit não achei nada parecido, apesar do Arduino usar uma variação do C....

0 0 Responder Compartilhar >



Silvano Vieira

5 anos atrás

Olá, boa tarde! Seu programa para caracteres especiais funcionou muito bem no display lcd! Como posso fazer o mesmo no display OLED, usando biblioteca Adafruit ou outra? Desde já, Obrigado pela atenção!

0 0 Responder Compartilhar >



Vinícius Willers

5 anos atrás editado

Boa tarde. Estou a alguns minutos tentando calcular para chegar no valor que você definiu de 7.6mA de consumo, colocando um resistor de 220 ohm e a corrente de 5V do pino 15 do display.

Pela lei de Ohm: $R = V / I$

Onde: $R = 5V / 0,0076A = 657,89$ ohm

Pode me indicar como fez o cálculo e como chegou no valor de 7,6mA (0,0076A) com 5v e usando 220ohm?

Estou para ligar esse display no arduino, porém estou com medo de queimá-lo por não saber exato qual resistor usar e qual a corrente que ele vai permitir passar.

Estou pensando em usar 120ohm no resistor, daí ficaria:

$R = 5v / 120ohm = 41,6mA (0,0416A)$.

Em tempo, adicionando mais uma informação:

$I = V / R$, então : $I = 5v / 220 ohm = \sim 22mA (0.022A)$

O cálculo inicial que você fez do consumo já estava considerando o uso de resistor então?

Obrigado desde já

0 0 Responder Compartilhar >



Jose Gustavo Abreu Murta

→ Vinícius Willers



5 anos atrás

Boa noite Vinícius,
Eu não calculei o valor da corrente! Eu medi com amperímetro quando eu inseri o resistor de 220 ohms em série com o LED do display. O display já tem um resistor interno em série com o LED. Pode ligar o pino LED+ diretamente no + 5V como escrevi no tutorial (veja o diagrama).
Atenciosamente,
Gustavo Murta

0 0 Responder Compartilhar >



Vinícius Willers

→ Jose Gustavo Abreu Murta

5 anos atrás

Obrigado pela rápida resposta Jose. Agora entendi perfeitamente. Eu estava receoso que não teria um resistor interno e eu ia fritar o display hehe. Muito obrigado, abraço!

0 0 Responder Compartilhar >



Laryssa Barbalho

6 anos atrás

Olá gostaria muito da sua ajuda, estou construindo um protótipo com arduino para meu tcc, pretendo através desse protótipo ensinar conteúdos básicos da disciplina da inglês, queria tirar algumas dúvidas com o senhor. Aguardo um retorno

0 0 Responder Compartilhar >



Jose Gustavo Abreu Murta

→ Laryssa Barbalho

6 anos atrás

Bom dia Laryssa,
Qual seria a sua dúvida?
Atenciosamente,
Gustavo Murta

0 0 Responder Compartilhar >



Lindonil Borracheiromóvel de C

7 anos atrás

muito obrigado pela aula aqui no seu blog.

0 0 Responder Compartilhar >



Jose Gustavo Abreu Murta

→ Lindonil Borracheiromóvel de C

7 anos atrás

Obrigado pelos elogios.
Atenciosamente,
Gustavo Murta

0 0 Responder Compartilhar >

Inscreve-se

Privacidade

Política de Proteção de Dados

Conheça a Metodologia Eletrogate e Lecione um [Curso de Robótica nas Escolas](#) da sua Região!

IOT

Monitoramento do Nível de Reservatórios de Água

Dalton.Menezes 27 de março de 2025



Este projeto tem como objetivo implementar o Monitoramento do Nível de Reservatórios de Água (Caixa D'Água ou Cisterna), através de estações de coleta (N) e uma estação central para visualização dos dados coletados pelas estações coletoras,... comunicando através de uma Rede Lora.

SENORES

Sensor Infravermelho E18-D80NK com Arduino

[Abraão da Silva](#) 7 de março de 2025



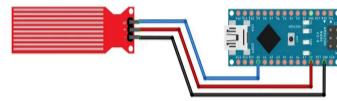
O E18-D80NK é um sensor de proximidade infravermelho que utiliza luz infravermelha (IR) para detectar objetos. Vamos explorar o sensor infravermelho E18-D80NK, um dispositivo utilizado para detectar objetos sem a necessidade de contato físico.

SENORES

Monitorando o Nível de Água com um Sensor de Chuva e o Plotter Serial do Arduino

[Eletrogate](#) 28 de fevereiro de 2025

Atualizado em: 07 mar 2025

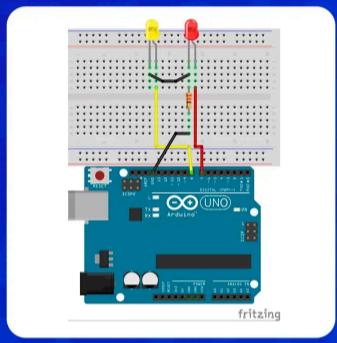


Hoje, vamos aprender como utilizar o Módulo Sensor de Chuva / Nível de Água com um microcontrolador Arduino para monitorar o nível de água e visualizar os dados no Plotter Serial.

PROJETOS

Gerador de gráficos de funções lineares e quadráticas com Arduino

[Eletrogate](#) 19 de fevereiro de 2025



Aprenda como montar um gerador de gráficos de funções lineares e quadráticas com Arduino e também sobre comunicação bidirecional entre Arduino-Computador.

Cadastre-se e fique por dentro de novidades!

Nome *

Email *

Eu aceito (Política de Privacidade)

Cadastrar

BLOG

Nos acompanhe



Facebook



Instagram



Youtube



Twitter



Pinterest

© ELETROGATE 2025 - Todos os direitos reservados. Termos de uso e Política de privacidade.