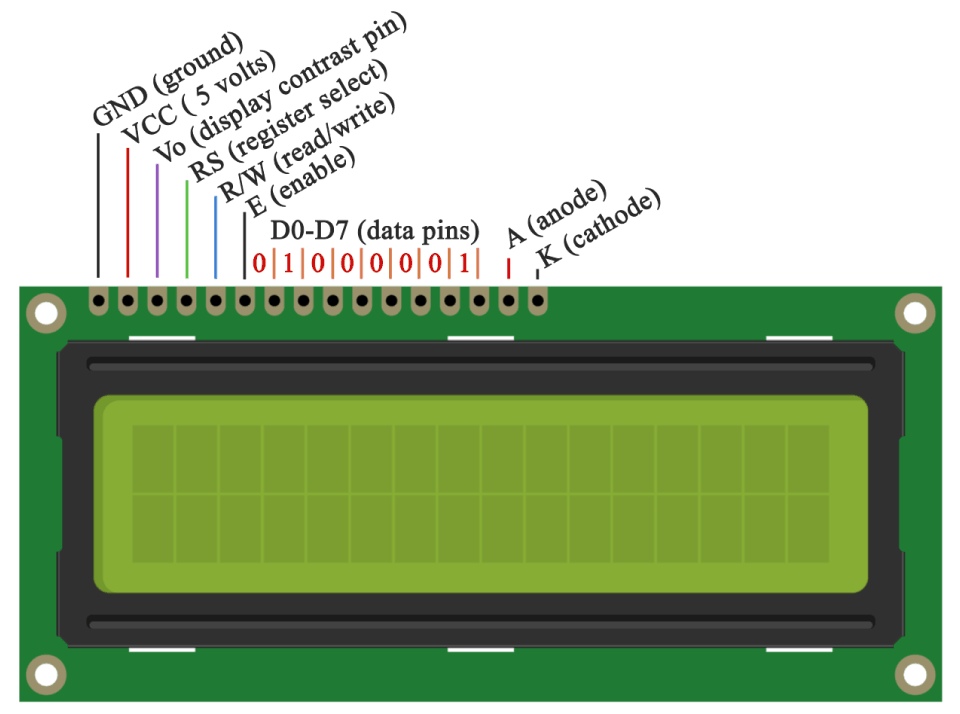
Documentação Liquid Crystal



|  |  |  |
| --- | --- | --- |
| Pino | Função | Descrição |
| 1 | Alimentação | GND |
| 2 | Alimentação | VCC |
| 3 | V0 | Pino para ajuste de contraste |
| 4 | RS | 1 = Dado, 0 = Instrução |
| 5 | R/W | 1 = Leitura, 0 = Escrita |
| 6 | E(Chip select) | 1 = Habilita, 0 = Desabilita |
| 7 | B0 | Pino de dados(bit mais significativo) |
| 8 | B1 | Pino de dados |
| 9 | B2 | Pino de dados |
| 10 | B3 | Pino de dados |
| 11 | B4 | Pino de dados |
| 12 | B5 | Pino de dados |
| 13 | B6 | Pino de dados |
| 14 | B7 | Pino de dados |
| 15 | A | Ânodo do LED de backlight |
| 16 | K | Cátodo do LED de backlight |

* LiquidCrystal()

Cria uma variável do tipo LiquidCrystal, quando instanciado, você pode escolher usar 4 ou 8 pinos do display, escolhendo colocar os valores no parâmetro da função ou não.

Sintaxe:

*LiquidCrystal(rs, enable, d4, d5, d6, d7)*

*LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)*

*LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)*

*LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)*

Onde:

**rs**: o número do pino que está conectado no pino RS no LCD

**rw**: número do pino que está conectado ao pino RW no LCD (opcional)

**enable**: número do pino que está conectado ao pino enable do LCD

**d0, d1, d2, d3, d4, d5, d6, d7**: número dos pinos dos dados do LCD. d0, d1, d2 e d3 são pinos opcionais, se eles não forem utilizados, os dados controlados serão somente nos pinos d4, d5, d6 e d7

**Funções básicas**

* begin() Inicializa a interface do LCD, esta função deve ser chamada antes que qualquer outra da biblioteca, pois ela inicializa o display depois de ele instanciado.

*Sintaxe: lcd.begin(16, 2)*

* clear() Limpa o lcd e seta o cursor para o início.

*Sintaxe: lcd.clear();*

* home() Seta o cursor para a posição inicial do display.

*Sintaxe: lcd.home()*

* setCursor() Caminha o cursor para a posição desejada.

*Sintaxe: lcd.setCursor(col, linha)*

* write() Printa um caractere ou variável no display.

*Sintaxe: lcd.write(dado)*

* print() Escreve no display uma sequência de caracteres, o dado pode ser do tipo char, inteiro, byte, long ou string.

Sintaxe: *lcd.write(“Hello World!”)*

*lcd.write(12345)*

* cursor() Habilita a localização atual do cursor, ele pisca abaixo do local onde será escrito o próximo caractere.

*Sintaxe: lcd.cursor()*

* noCursor() Desabilita a localização atual do cursor (ver função cursor)

*Sintaxe:lcd.noCursor()*

* blink() Se habilitada a função cursor, é possível visualizar ele piscando

*Sintaxe:lcd.blink()*

* noBlink() Se o cursor do LCD está piscando, ele é desativado.

*Sintaxe:lcd.noBlink()*

* display() Habilita o display após ser utilizada a função noDisplay()

*Sintaxe:lcd.display()*

* noDisplay() Desabilita o display LCD

*Sintaxe:lcd.noDisplay()*

* scrollDisplayLeft() Arrasta o texto e cursor da tela um espaço para a esquerda

*Sintaxe:lcd.scrollDisplayLeft()*

* scrollDisplayRight() Arrasta o texto e cursor da tela um espaço para a direita

*Sintaxe:lcd.scrollDisplayRight()*

* autoscroll() Se o texto está sendo escrito da esquerda para a direita, ele arrasta o texto automaticamente para a esquerda, adaptando o visor com novos textos.

*Sintaxe:lcd.autoScroll()*

* noAutoscroll() Desabilita a função autoscroll,

*Sintaxe:lcd.noAutoscroll()*

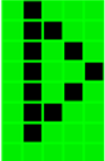
* leftToRight() Coloca a direção do texto escrito da esquerda para a direita por padrão, quando colocado um texto no display, ele é exibido nessa direção.

*Sintaxe:lcd.leftToRight()*

* rightToLeft() Coloca a direção do texto escrito da dirita para a esquerda por padrão, quando colocado um texto no display, ele é exibido nessa direção.

*Sintaxe:lcd.rightToLeft()*

* createChar() Cria um caracter especial no LCD, são suportados caracteres são colocados em um molde de 5x8 píxels, para fazer o preenchimento dele, é utilizado um vetor de 8 posições (quantidade de linhas do nosso molde) e cada linha temos os 5 espaços que são representados por bits 0 e 1.



byte numeros[8] = {

B01000,

B01100,

B01010,

B01001,

B01100,

B01000,

B00000,

};

lcd.createChar(posicao, numeros)

posicao - posicao do caracter especial salva na memória, é possível salvar até

Para mostrar no display o caracter especial, utilizamos o lcd.write(byte(posicao)), caso seja digitado um valor diferente do gravado, ele pegará o valor que é guardado na tabela 01 do display de fábrica.



Exemplo de Código para visor LCD:

#include <LiquidCrystal.h>

**//Exemplo LiquidCrystal**

LiquidCrystal lcd(13, 12, 5, 4, 3, 2);

void setup() {

lcd.begin(16, 2);

}

//efeito de piscar

void efeito1(){

lcd.noDisplay();

delay(500);

lcd.display();

delay(500);

}

//efeito de andar para esquerda

void efeito2(){

lcd.scrollDisplayLeft();

delay(350);

}

//anda o display para direita

void efeito3(){

lcd.scrollDisplayRight();

delay(350);

}

void loop() {

lcd.setCursor(0, 0);

lcd.print("Hello World");

efeito1();

}