

# PDM: Projeto para Dispositivos Móveis

## Aula 10: Utilizando o LCD

Breno Lisi Romano

<http://sites.google.com/site/blromano>

Instituto Federal de São Paulo – IFSP São João da Boa Vista  
Especialização em Desenvolvimento para Dispositivos Móveis

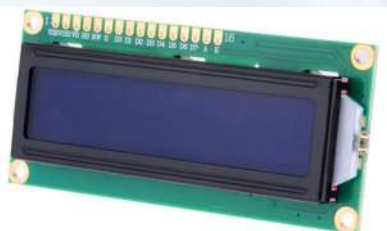


INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SÃO PAULO  
Campus São João da Boa Vista

Instituto Federal de São Paulo – IFSP São João da Boa Vista

## Introdução (1)

- Nesta aula, será utilizado o Display LCD (*Liquid Crystal Display*) 16×2
- Display muito comum com controlador HD44780, que se adapta aos mais diversos projetos, podendo ser usado com vários modelos de placas e microcontroladores como Arduino, Raspberry, PIC



## Introdução (2)

- Esse display LCD tem 16 colunas e 2 linhas, com *backlight* (luz de fundo) azul e letras na cor branca
- Para conexão, são 16 pinos, dos quais usa-se 12 para uma conexão básica, já incluindo as conexões de alimentação (pinos 1 e 2), *backlight* (pinos 15 e 16) e contraste (pino 3)

## Introdução (3)

- Excelente link de referência: <https://www.arduino.cc/en/Reference/LiquidCrystal>
- A interface do LCD consiste dos seguintes pinos:
  - Um pino RS (*Register Select*) que controla onde na memória do LCD pode-se escrever informações
    - Pode-se selecionar: o Registrador de Dados – o que será escrito na Tela; Registrador de Instruções – onde o controlador do LCD irá buscar por instruções para realizar
  - Um pino (*Read/Write*) que seleciona o modo de escrita ou leitura
  - Um pino E (*Enable*) para habilitar a escrita no registrador
  - 8 pinos de dados (D0 -D7): Os estados destes pinos (high or low) são bits escritos ou lidos no registrador
  - Pino do Constraste do Display (Vo), Pinos de Energia e Terra (+5V and Gnd) e os pinos da cor de fundo do LED (LED Backlight - Bklt+ and BKlt-) utilizados para ligar o LCD, controlar o contraste do display e ligar/desligar o LED backlight, respectivamente

## ***Problem Based Learnd - PBL 01***

- **Problema:** Utilizar alguma funções básicas do LCD 16x2, como posicionar cursor, imprimir texto, mover texto

## **Informações Necessárias (*Background*) (1)**

- Os LCDs podem ser controlados de dois modos: 4-bit or 8-bit
  - O modo de 4-bit necessita de 7 pinos I/O pins do Arduino enquanto que o modo de 8-bit necessita de 11 pinos. Para mostrar textos no display LCD, pode-se utilizar o modo de 4-bits
  - Principal vantagem de se utilizar 8-bits: aumentar a velocidade da transmissão de dados entre LCD/Arduino
  - Na conexão do display ao Arduino Uno, deve-se utilizar apenas 4 pinos de dados (pinos digitais 2, 3, 4 e 5), e 2 pinos de controle (pinos digitais 11 e 12)
- Para o ajuste do contraste, pode-se utilizar um potenciômetro de 10k

## Informações Necessárias (*Background*) (2)

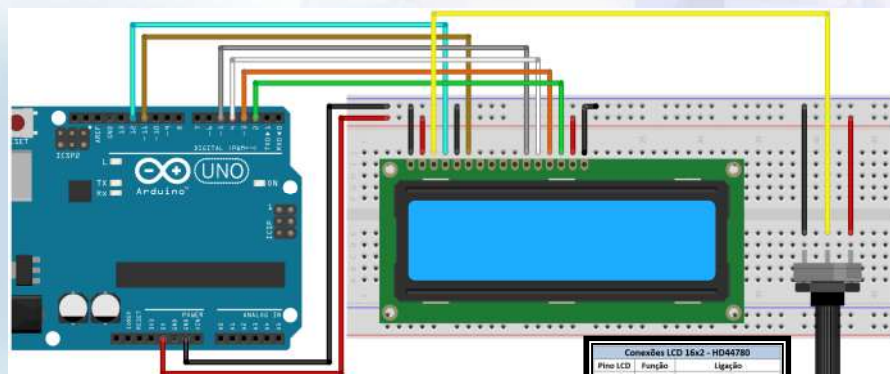
Pino	Símbolo	Função
1	VSS	GND(Alimentação)
2	VDD	5V(Alimentação)
3	V0	Ajuste de Contraste
4	RS	Habilita/Desabilita Seletor de Registrador
5	R/W	Leitura/Escrita
6	E	Habilita Escrita no LCD
7	DB0	Dado
8	DB1	Dado
9	DB2	Dado
10	DB3	Dado
11	DB4	Dado
12	DB5	Dado
13	DB6	Dado
14	DB7	Dado
15	A	5V(Backlight)
16	K	GND(BackLight)

Conexões LCD 16x2 - HD44780		
Pino LCD	Função	Ligação
1	Vss	GND
2	Vdd	Vcc 5V
3	V0	Pino central potenciômetro
4	RS	Pino 12 Arduino
5	RW	GND
6	E	Pino 11 Arduino
7	D0	Não conectado
8	D1	Não conectado
9	D2	Não conectado
10	D3	Não conectado
11	D4	Pino 5 Arduino
12	D5	Pino 4 Arduino
13	D6	Pino 3 Arduino
14	D7	Pino 2 Arduino
15	A	Vcc 5V
16	K	GND

## Componentes Utilizados

- Arduino UNO / Arduino MEGA
- Cabo USB
- Protoboard
- Potenciômetro 10k
- LCD 16x2

## Construindo o Circuito no Protoboard



Os pinos de dados (Data Pins) de 0 a 3 não são usados, pois será utilizado o modo de 4 bits

Pino LCD	Função	Ligação
1	Vss	GND
2	Vdd	Vcc 5V
3	VO	Pino central potenciômetro
4	R/S	Pino 12 Arduino
5	R/W	GND
6	E	Pino 11 Arduino
7	D0	Não conectado
8	D1	Não conectado
9	D2	Não conectado
10	D3	Não conectado
11	D4	Pino 5 Arduino
12	D5	Pino 6 Arduino
13	D6	Pino 9 Arduino
14	D7	Pino 2 Arduino
15	A	Vcc 5V
16	K	GND

## Código Fonte para Solução (1)

```

1 //Carrega a biblioteca LiquidCrystal
2 #include <LiquidCrystal.h>
3
4 //Define os pinos que serão utilizados para ligação ao display
5 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
6
7 void setup()
8 {
9     //Define o número de colunas e linhas do LCD
10    lcd.begin(16, 2);
11 }
12

```

## Código Fonte para Solução (2)

```
13 void loop()
14 {
15     //Limpa a tela
16     lcd.clear();
17
18     //Posiciona o cursor na coluna 3, linha 0;
19     lcd.setCursor(3, 0);
20
21     //Envia o texto entre aspas para o LCD
22     lcd.print("HELLO WORD");
23     lcd.setCursor(3, 1);
24     lcd.print(" LCD 16x2");
25     delay(5000);
26
27     //Rolagem para a esquerda
28     for (int posicao = 0; posicao < 3; posicao++)
29     {
30         lcd.scrollDisplayLeft();
31         delay(300);
32     }
33
34     //Rolagem para a direita
35     for (int posicao = 0; posicao < 6; posicao++)
36     {
37         lcd.scrollDisplayRight();
38         delay(300);
39     }
40 }
```

## Verificação e Simulação do Código Desenvolvido

- Embarcar o código-fonte na aplicação e verificar o resultado no Display LCD 16x2

## Problem Based Learnd - PBL 02

- **Problema:** Parametrizar as funções disponíveis no Display LCD 16x2 e mostrar o funcionamento das funções biblioteca “LiquidCrystal.h”
  - basicPrintDemo()
  - displayOnOffDemo()
  - setCursorDemo()
  - scrollLeftDemo()
  - scrollRightDemo()
  - cursorDemo()
  - createGlyphDemo()

## Informações Necessárias (*Background*) (1)

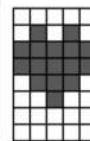
- Criando caracteres especiais no LCD:
  - Com os displays baseados no HD44780, pode-se criar até 8 caracteres customizados
  - Primeiro, deve-se criar um vetor de 8 bytes que representará o nosso caracter

```
byte heart[8] = { 0b00000,
```

```
0b01010,
0b11111,
0b11111,
0b01110,
0b00100,
0b00000,
0b00000 };
```

Entendeu? Ai vai uma melhor então:

```
byte heart[8] = {
  0b00000,
  0b01010,
  0b11111,
  0b11111,
  0b01110,
  0b00100,
  0b00000,
  0b00000 };
```

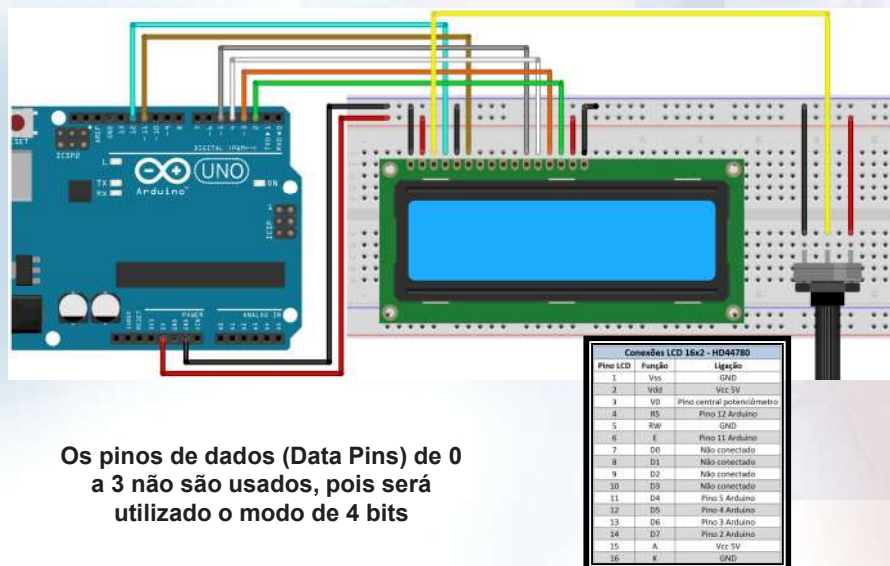


```
lcd.createChar(1, heart); // Grava o caracter heart na posição 1.
```

```
lcd.write(1); // Escreve o caracter da posição 1 no display.
```



## Construindo o Circuito no Protoboard



## Código Fonte para Solução (1)

```

1 #include <LiquidCrystal.h>
2
3 // initialize the library with the numbers of the interface pins
4 LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // create an lcd object and assign the pins
5
6 void setup() {
7     lcd.begin(16, 2); // Set the display to 16 columns and 2 rows
8 }
9
10 void loop() {
11     // run the 7 demo routines
12     basicPrintDemo();
13     displayOnOffDemo();
14     setCursorDemo();
15     scrollLeftDemo();
16     scrollRightDemo();
17     cursorDemo();
18     createGlyphDemo();
19 }
20
21 void basicPrintDemo() {
22     lcd.clear(); // Clear the display
23     lcd.print("Basic Print"); // print some text
24     delay(2000);
25 }

```



## Código Fonte para Solução (2)

```

27 void displayOnOffDemo() {
28     lcd.clear(); // Clear the display
29     lcd.print("Display On/Off"); // print some text
30     for(int x=0; x < 3; x++) { // loop 3 times
31         lcd.noDisplay(); // turn display off
32         delay(1000);
33         lcd.display(); // turn it back on again
34         delay(1000);
35     }
36 }
37
38 void setCursorDemo() {
39     lcd.clear(); // Clear the display
40     lcd.print("SetCursor Demo"); // print some text
41     delay(1000);
42     lcd.clear(); // Clear the display
43     lcd.setCursor(5,0); // cursor at column 5 row 0
44     lcd.print("5,0");
45     delay(2000);
46     lcd.setCursor(10,1); // cursor at column 10 row 1
47     lcd.print("10,1");
48     delay(2000);
49     lcd.setCursor(3,1); // cursor at column 3 row 1
50     lcd.print("3,1");
51     delay(2000);
52 }

```

## Código Fonte para Solução (3)

```

54 void scrollLeftDemo() {
55     lcd.clear(); // Clear the display
56     lcd.print("Scroll Left Demo");
57     delay(1000);
58     lcd.clear(); // Clear the display
59     lcd.setCursor(7,0);
60     lcd.print("Beginning");
61     lcd.setCursor(9,1);
62     lcd.print("Arduino");
63     delay(1000);
64     for(int x=0; x<16; x++) {
65         lcd.scrollDisplayLeft(); // scroll display left 16 times
66         delay(250);
67     }
68 }
69
70 void scrollRightDemo() {
71     lcd.clear(); // Clear the display
72     lcd.print("Scroll Right");
73     lcd.setCursor(0,1);
74     lcd.print("Demo");
75     delay(1000);
76     lcd.clear(); // Clear the display
77     lcd.print("Beginning");
78     lcd.setCursor(0,1);
79     lcd.print("Arduino");
80     delay(1000);
81     for(int x=0; x<16; x++) {
82         lcd.scrollDisplayRight(); // scroll display right 16 times
83         delay(250);
84     }
85 }

```

## Código Fonte para Solução (4)

```

87 void cursorDemo() {
88     lcd.clear(); // Clear the display
89     lcd.cursor(); // Enable cursor visible
90     lcd.print("Cursor On");
91     delay(3000);
92     lcd.clear(); // Clear the display
93     lcd.noCursor(); // cursor invisible
94     lcd.print("Cursor Off");
95     delay(3000);
96     lcd.clear(); // Clear the display
97     lcd.cursor(); // cursor visible
98     lcd.blink(); // cursor blinking
99     lcd.print("Cursor Blink On");
100    delay(3000);
101    lcd.noCursor(); // cursor invisible
102    lcd.noBlink(); // blink off
103 }

```

## Código Fonte para Solução (5)

```

106 void createGlyphDemo() {
107     lcd.clear();
108
109     byte happy[8] = { // create byte array with happy face
110         B00000, B00000,
111         B10001, B00000,
112         B10001, B01110,
113         B00000, B00000};
114
115     byte sad[8] = { // create byte array with sad face
116         B00000, B00000,
117         B10001, B00000,
118         B01110, B10001,
119         B00000, B00000};
120     lcd.createChar(1, happy); // create custom character 0
121     lcd.createChar(2, sad); // create custom character 1
122
123     for(int x=0; x<5; x++) { // loop animation 5 times
124         lcd.setCursor(8,0);
125         lcd.write(1); // write custom char 0
126         delay(1000);
127         lcd.setCursor(8,0);
128         lcd.write(2); // write custom char 1
129         delay(1000);
130     }
131 }

```

Instituto Federal de São Paulo – IFSP São João da Boa Vista

## Verificação e Simulação do Código Desenvolvido

- Embarcar o código-fonte na aplicação e verificar o resultado no Display LCD 16x2

Instituto Federal de São Paulo – IFSP São João da Boa Vista

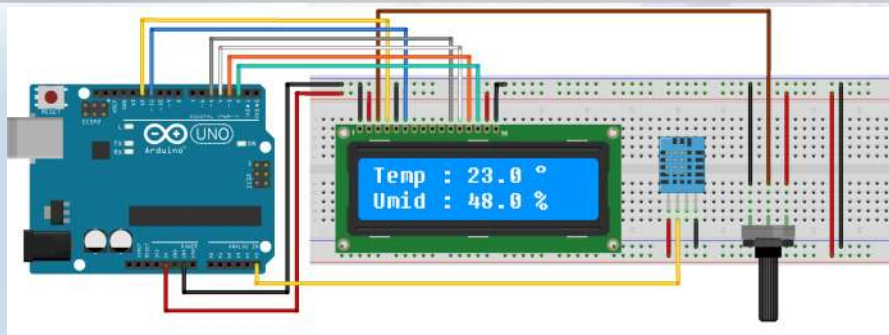
## *Problem Based Learnd - PBL 03*

- **Problema:** Mostrar os valores medidos de Temperatura e Umidade no Display LCD 16x2

## Componentes Utilizados

- Arduino UNO / Arduino MEGA
- Cabo USB
- Protoboard
- Potenciômetro 10k
- LCD 16x2
- Sensor de Temperatura/Umididade DHT11

## Construindo o Circuito no Protoboard



- Pino 4 (RS) ligado ao pino 12 do Arduino
- Pino 6 (E) ligado ao pino 11 do Arduino
- Pino 11 (D4) ligado ao pino 5 do Arduino
- Pino 12 (D5) ligado ao pino 4 do Arduino
- Pino 13 (D6) ligado ao pino 3 do Arduino
- Pino 14 (D7) ligado ao pino 2 do Arduino
- Pino 3 do display será ligado ao pino central de um potenciômetro de 10K, que tem a função de regular o contraste.
- As demais ligações são feitas ao GND (pinos 1, 5 e 16) e aos 5v do Arduino (pinos 2 e 15), e qualquer inversão pode impedir a exibição dos caracteres

## Código Fonte para Solução (1)

```

3 #include <LiquidCrystal.h> //Carrega a biblioteca LCD
4 #include <DHT.h> //Carrega a biblioteca DHT
5
6 //Define a ligação ao pino de dados do sensor
7 #define DHTPIN A5
8
9 //Define o tipo de sensor DHT utilizado
10 #define DHTTYPE DHT11
11
12 DHT dht(DHTPIN, DHTTYPE);
13
14 //Define os pinos que serão ligados ao LCD
15 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
16
17 //Array simbolo grau
18 byte grau[8] = { B00001100,
19                  B00010010,
20                  B00010010,
21                  B00001100,
22                  B00000000,
23                  B00000000,
24                  B00000000,
25                  B00000000, };
26

```

## Código Fonte para Solução (2)

```

void setup()
{
  Serial.begin(9600); //Inicializa a serial
  lcd.begin(16,2); //Inicializa LCD
  lcd.clear(); //limpa o LCD
  //Cria o caractere customizado com o simbolo do grau
  lcd.createChar(0, grau);
}

void loop()
{
  float h = dht.readHumidity(); //Le o valor da umidade
  float t = dht.readTemperature(); //Le o valor da temperatura
  lcd.setCursor(0,0);
  lcd.print("Temp : ");
  lcd.setCursor(7,0);
  lcd.print(t,1);
  lcd.setCursor(12,0);

  //Mostra o simbolo do grau formado pelo array
  lcd.write((byte)0);

  //Mostra o simbolo do grau quadrado
  //lcd.print((char)223);

  lcd.setCursor(0,1);
  lcd.print("Umidade : ");
  lcd.setCursor(7,1);
  lcd.print(h,1);
  lcd.setCursor(12,1);
  lcd.print("%");

  //Intervalo recomendado para leitura do sensor
  delay(2000);
}

```

*lcd.print(data, BASE)*

**Parâmetros:**

- data: dado a ser impresso (char, byte, int, long, ou string)
- BASE (optional): a base para imprimir números: BIN para binário (base 2), DEC para decimal (base 10), OCT para octal (base 8), HEX para hexadecimal (base 16).

Instituto Federal de São Paulo – IFSP São João da Boa Vista

## Verificação e Simulação do Código Desenvolvido

- Embarcar o código-fonte na aplicação e verificar o resultado no Display LCD 16x2

Instituto Federal de São Paulo – IFSP São João da Boa Vista

## Miniprojeto 19

- Acionar a leitura de Temperatura, Umidade, Luminosidade ou da Distância utilizando um Aplicativo Android integrado por Bluetooth com um Circuito Arduino e mostrar o resultado das medidas solicitadas e lidas no Display LCD 16x2
  - Mostrar no LCD apenas a Leitura Solicitada

## Miniprojeto 20

- Simular o registro de presenças de um aluno em um Evento utilizando um Aplicativo Android, no qual ele lê um QR Code que identifica o evento e confirma a entrada do aluno mostrando o Nome e o Prontuário do mesmo em um Display LCD 16x2 presente em um circuito com Arduino, integrado com o Aplicativo Android por Bluetooth

## PDM: Projeto para Dispositivos Móveis

### Aula 10: Utilizando o LCD

Breno Lisi Romano

**Obrigado!**

Instituto Federal de São Paulo – IFSP São João da Boa Vista  
Especialização em Desenvolvimento para Dispositivos Móveis



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SÃO PAULO  
Campus São João da Boa Vista