



Co-funded by the  
Erasmus+ Programme  
of the European Union



# UNIDADE 6

## MONITORES LCD (LIQUID CRISTAL DISPLAY)

# Objetivo e Conteúdo da Unidade 6

## *Objetivo*

Analisar o monitor LCD como um periférico de output: permite visualizar qualquer tipo de informação de output incluindo números, letras e símbolos

## *Conteúdo*

- Explicação sobre conceitos básicos de monitores LCD
- Explicação sobre **conjunto de caracteres e caracteres gráficos**
- Apresentação da biblioteca **liquidCrystal.h**

# MONITORES LCD

- Este é um periférico de output que não só mostra números mas também todo o tipo de caracteres, textos, símbolos e até mesmo gráficos simples
- Vai utilizar um monitor LCD de 2 x 16 LCD



- O monitor LCD é um periférico digital. Os seus sinais podem ser ligados diretamente aos pins de input e output do controlador.
- Dividem-se em três grupos:
  - ✓ Abastecimento
  - ✓ Controlo
  - ✓ Dados

# MONITORES LCD

Observe a tabela com a descrição de cada um dos pins

PIN Nº	NOME	TIPO	DESCRIÇÃO
1	Vss	Fornecimento de Voltagem	Abastecimento de energia terra (0 V)
2	Vdd	Fornecimento de Voltagem	Abastecimento de energia positiva +5 Vcc
3	VLC	Fornecimento de Voltagem	Ajuste de contraste: voltagens a alternar entre 0 e +5 Vcc.
4	RS	Input	Output do controlador Arduino. Seleciona entre instruções e dados: RS=0 Arduino transfere instruções RS=1 Arduino transfere dados (códigos ASCII)
5	R/W	Input	Output do Arduino. Controla a leitura e a escrita: R/W=0 Arduino escreve dados no ecrã LCD R/W=1 Arduino lê dados do ecrã LCD
6	E	Input	Output do Arduino. Viabiliza o ecrã: E=0 ecrã LCD desativado (impedimento elevado) E=1 ecrã LCD ativado
7-14	DB0:DB7	Input/Output	Linhas bus de dados e instruções. O Arduino transfere instruções ou dados para o ecrã de acordo com o sinal RS. DB0: linhas DB7 são utilizadas com uma interface de 8 bits DB4: linhas DB7 são utilizadas com uma interface de 4 bits
15	L+	Fornecimento de Voltagem	Voltagem positiva para a iluminação de background (+5 Vcc)
16	L-	Fornecimento de Voltagem	Voltagem negativa para a iluminação de background (0 V)

# O GRUPO DE CARATERES

- A comunicação entre o Arduino e os monitores LCD efetua-se basicamente através dos pins digitais DB0-DB7
- Envia os códigos dos caracteres ASCII que pretende mostrar. São códigos de 8 bits

## **A memória ROM interna contém a definição de cada um**

- Se utilizar uma interface de “8 bit”, o Arduino precisa apenas de uma única transferência para mostrar cada carater
- Uma interface de “4 bits” precisa de duas transferências para mostrar cada carater

# OS CARACTERES GRÁFICOS

- É possível criar um total de oito caracteres gráficos de 5 x 8 pontos ou “pixels”, cada caracter´ está numerado de 0 a 7 e precisa de um total de 8 bytes para ser definido.

**O ecrã LCD tem uma memória RAM interna denominada CGRAM para executar esta tarefa**

- Os caracteres gráficos definem-se ao inserirem-se bytes em posições sucessivas de memória CGRAM
- A CGRAM é uma memória volátil capaz de armazenar um total de 64 bytes

## Vamos estudar as funções mais representativas

### ➤ FUNÇÃO: LIQUIDCRYSTAL()

- ✓ Esta função cria um tipo “LiquidCrystal” variável e estabelece as ligações entre o monitor LCD e o controlador Arduino

```
LiquidCrystal var(RS,E,D4,D5,D6,D7);           //Para interface de 4 bits sem sinal R/W  
LiquidCrystal var(RS,RW,E,D4,D5,D6,D7);       //Para interface de 4 bits com sinal R/W  
LiquidCrystal var(RS,E,D0,D1,D2,D3,D4,D5,D6,D7); //Para interface de 8 bits sem sinal R/W  
LiquidCrystal var(RS,RW,E,D0,D1,D2,D3,D4,D5,D6,D7); //Para interface de 8 bits com sinal R/W
```

*var:* o nome da variável atribuída ao monitor LCD que vai controlar.

*RS:* o pin do Arduino ligado ao sinal do ecrã RS.

*RW:* o pin do Arduino ligado ao sinal do ecrã R/W (se for utilizado).

*E:* o pin do Arduino ligado ao sinal do ecrã E.

*D0-D7:* os pins do Arduino que estão ligados às linhas de dados do ecrã DB0-DB7. Se não houver pins adequados para DB0-DB3, assumimos a interface de 4 bits e só utilizamos sinais DB4-DB7.

## ➤ FUNÇÃO: BEGIN()

- ✓ Esta função inicia o ecrã LCD e atribui-lhe o número de linhas e o número de caracteres por linha de acordo com o modelo em causa

**var.begin(c,f);**

*var:* esta é a identificação que define o ecrã em causa (estabelecido no LyquidCrystal()).

*c:* número de colunas.

*f:* número de linhas.

## ➤ FUNÇÃO: SETCURSOR()

- ✓ Esta função posiciona o cursor do ecrã LCD como pretendido. Daí em diante, os caracteres anteriores são mostrados.

**var.begin(c,f);**

*var:* esta é a identificação que define o ecrã em causa (estabelecido no LyquidCrystal()).

*c:* número de colunas.

*f:* número de linhas.

## ➤ FUNÇÃO: HOME()

- ✓ Esta função atribui ao cursor na primeira posição o canto superior esquerdo do ecrã (posição 0 da linha 0)

**var.home();**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

## ➤ FUNÇÃO: CLEAR()

- ✓ Esta função limpa o ecrã LCD e localiza o cursor no canto superior esquerdo

**var.clear();**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

## ➤ FUNÇÃO: WRITE()

- ✓ Esta função escreve um carater na posição atual do cursor

**var.write(char);**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

*char:* carater a ser mostrado no ecrã.

## ➤ FUNÇÃO: PRINT ()

- ✓ Esta função imprime no ecrã LCD, começando da posição em que se encontra o cursor

**var.print(data,base);**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

*data:* Dados a serem impressos. Podem ser char, int, long, float ou string.

*base:* Opcional e mostra a base numérica desejada: BIN=Binária; DEC=Decimal (por defeito); OCT=Octal; HEX=Hexadecimal; ou N=nº em décimas para números floating-point (2 por defeito).

## ➤ FUNÇÃO: CURSOR()

- ✓ Esta função mostra o cursor no ecrã LCD na sua posição atual, na forma de um underscore (\_)

**var.cursor();**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

## ➤ FUNÇÃO: NOCURSOR()

- ✓ Esta função esconde o cursor LCD.

**var.noCursor();**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

## ➤ FUNÇÃO: BLINK()

- ✓ Esta função apresenta o cursor LCD cursor no ecrã na sua posição atual como um símbolo sólido intermitente (  ).

**var.blink();**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

## ➤ FUNÇÃO: NOBLINK()

- ✓ Esta função esconde o cursor intermitente sólido (  ).

**var.noblink();**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

## ➤ FUNÇÃO: DISPLAY()

- ✓ Esta função liga o ecrã LCD e recupera o conteúdo apresentado anteriormente nesse ecrã, antes do noDisplay() ser executado.

**var.display();**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

## ➤ FUNÇÃO: NODISPLAY()

- ✓ Esta função desliga o ecrã LCD sem perda de conteúdo, independentemente do conteúdo e da posição do professor.

**var.noDisplay();**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

## ➤ FUNÇÃO: SCROLLDISPLAYLEFT()

- ✓ Esta função muda, em qualquer momento, uma posição para a esquerda o conteúdo (o texto e a posição do cursor) mostrado no ecrã.

**var.scrollDisplayLeft();**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

## ➤ FUNÇÃO: SCROLLDISPLAYRIGHT()

- ✓ Esta função muda, em qualquer momento, uma posição para a direita o conteúdo (o texto e a posição do cursor) mostrado no ecrã.

**var.scrollDisplayRight();**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

## ➤ FUNÇÃO: LEFTTORIGHT()

- ✓ Esta função estabelece automaticamente a orientação da escrita do cursor no ecrã: da esquerda para a direita. Significa que os caracteres são registados da esquerda para a direita sem afetar os que já tenham sido escritos.

**var.LeftToRight ();**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

## ➤ FUNÇÃO: RIGHTTOLEFT()

- ✓ Esta função reverte a orientação em que o cursor escreve no ecrã: da direita para a esquerda. Isto significa que os caracteres são escritos da direita para a esquerda sem afetar os que já tenham sido escritos.

**var.RightToLeft();**

*var:* define o ecrã em causa (estabelecido no LyquidCrystal()).

## ➤ FUNÇÃO: AUTOSCROLL()

- ✓ Esta função ativa o deslizar ou movimento automático do ecrã. Cada vez que um carater é enviado para o ecrã, esta função apresenta-o e move o restante conteúdo. Se a direção for da esquerda para a direita (`leftToRight()`), o conteúdo é movimentado para a esquerda. Se a direção no momento for da direita para a esquerda (`rightToLeft()`), o conteúdo movimenta-se para a direita.

**`var.autoscroll();`**

*var:* define o ecrã em causa (estabelecido em `LyquidCrystal()`).

## ➤ FUNÇÃO: NOAUTOSCROLL ()

- ✓ Esta função desativa o deslizar ou movimento automático do ecrã.

**`var.noAutoscroll();`**

*var:* define o ecrã em causa (estabelecido em `LyquidCrystal()`).



Co-funded by the  
Erasmus+ Programme  
of the European Union



# UNIDADE 6

## Monitores LCD (Liquid Cristal Display)

### Obrigado!

