

5197 - Sistema Digitais

Bacharelado de Informática

UEM – DIN - Prof. Elvio

2016

Roteiro

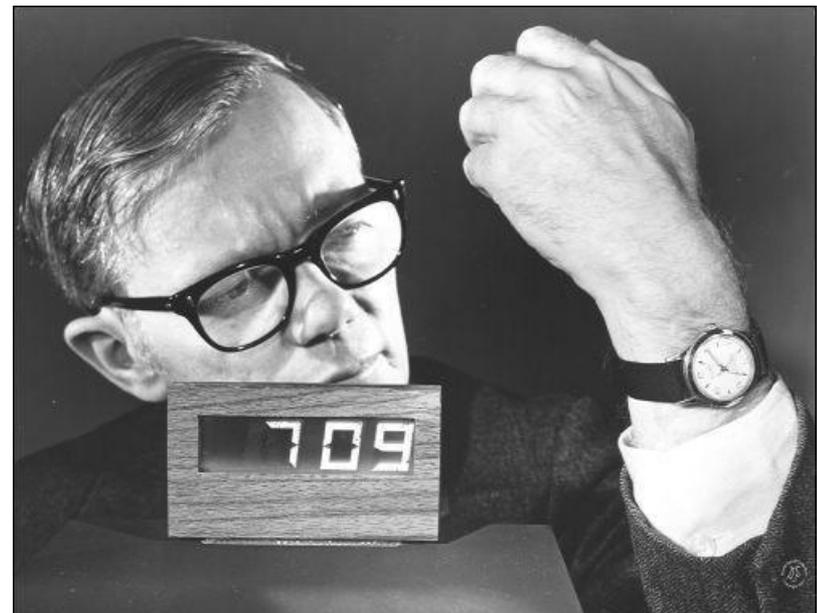
- LCD

O que é LCD?

- LCD: do inglês *liquid crystal display* ou display de cristal líquido
- Tecnologia tem evoluído significativamente nos últimos anos
- Resultado:
 - Melhora na qualidade dos painéis de LCD
 - Queda dos preços
- Resultado:
 - LCDs são encontrados hoje em produtos tão pequenos como relógios de pulso e telefones celulares e tão grandes como televisores de 60 polegadas
- O que acontece?
 - Os cristais no LCD se retorcem com a aplicação de uma tensão elétrica
 - Esta torção permite ou impede a passagem de luz pelo cristal, criando zonas mais claras e escuras, isto é, desenhando figuras na tela

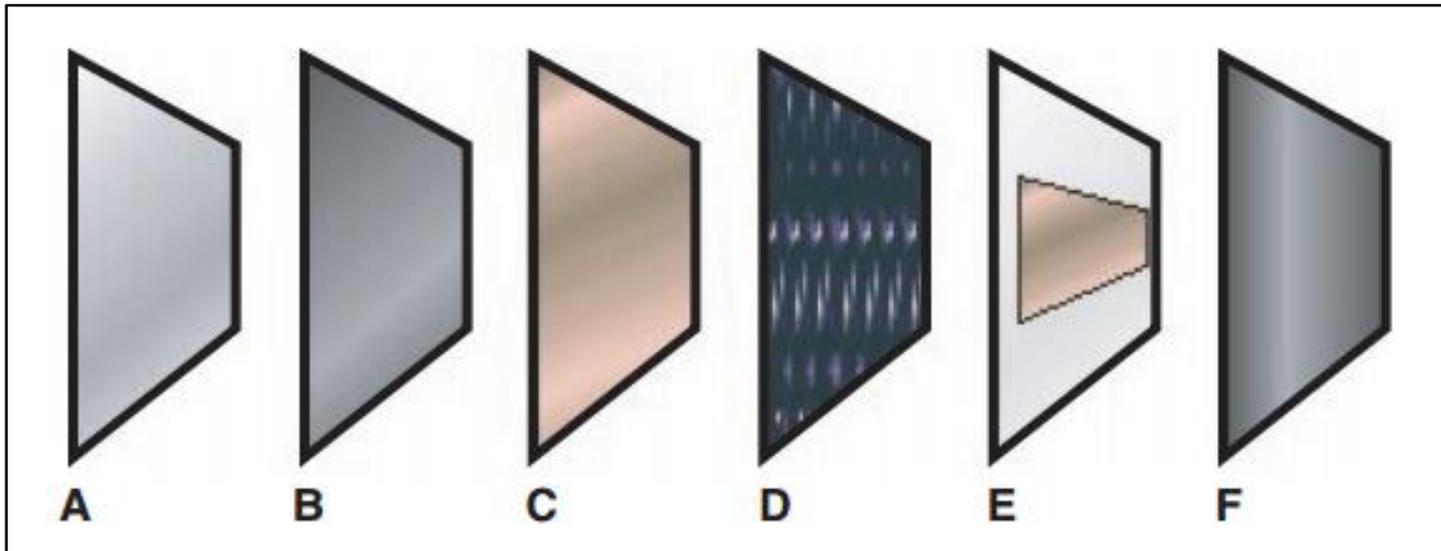
Histórico do LCD

- 1888: Friedrich Reinitzer descobre a natureza do líquido cristalino de colesterol extraído de cenoura
- Vários outros pesquisadores trabalham com cristais líquidos em diversos experimentos
- 1962: Richard Williams, da RCA, descobriu algumas características electro-ópticas interessantes de cristais líquidos, criando faixas em uma fina camada de material através da aplicação de uma tensão
- 1964: George Heilmeyer e equipe, da RCA, construíram o primeiro display de cristal líquido operacional
- RCA, em crise financeira, não explora comercialmente a invenção
- Década de 1970: cristais líquidos começam a ser usados em relógios de pulso
- 4º trimestre de 2007: venda de televisores LCD superou a de CRT em nível mundial



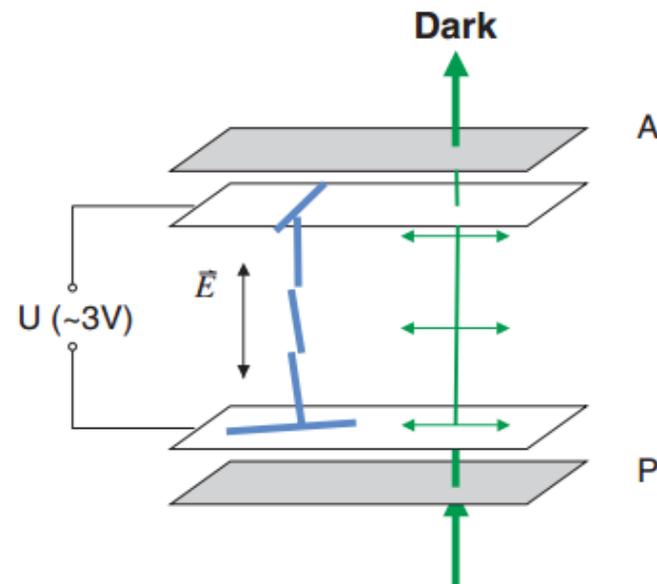
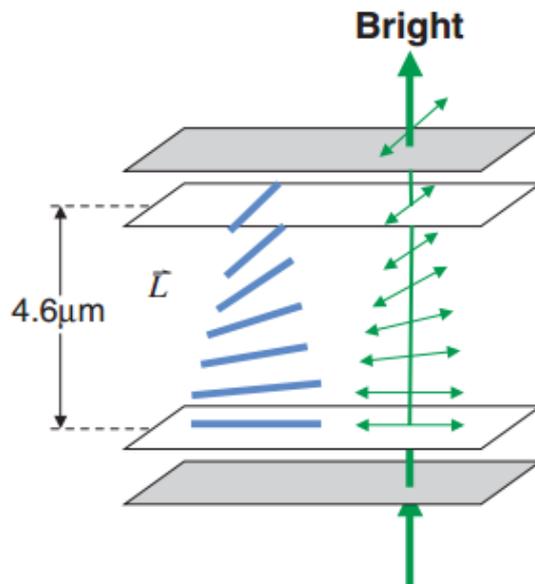
Como é construído?

- Um painel de LCD simples é composto de:
 - espelho na parte de trás (A)
 - vidro e filme polarizado (B)
 - eletrodo comum (C)
 - cristal líquido (D)
 - Vidro (E), com um eletrodo transparente no formato desejado
 - filme polarizado (F), com polarização ortogonal à (B)



Como é construído?

- Aplicação de tensão no eletrodo (E) permite ou impede a passagem de luz
- Cristais líquidos não emitem luz
- **LCD reflexivo**: mais baratos, possui espelho no fundo e apenas reflete a luz externa
 - Funcionam melhor em ambiente bem iluminado
- **LCD *back lit***: possui iluminação colocada acima, ao lado ou no fundo
 - Funcionam melhor em ambientes pouco iluminados

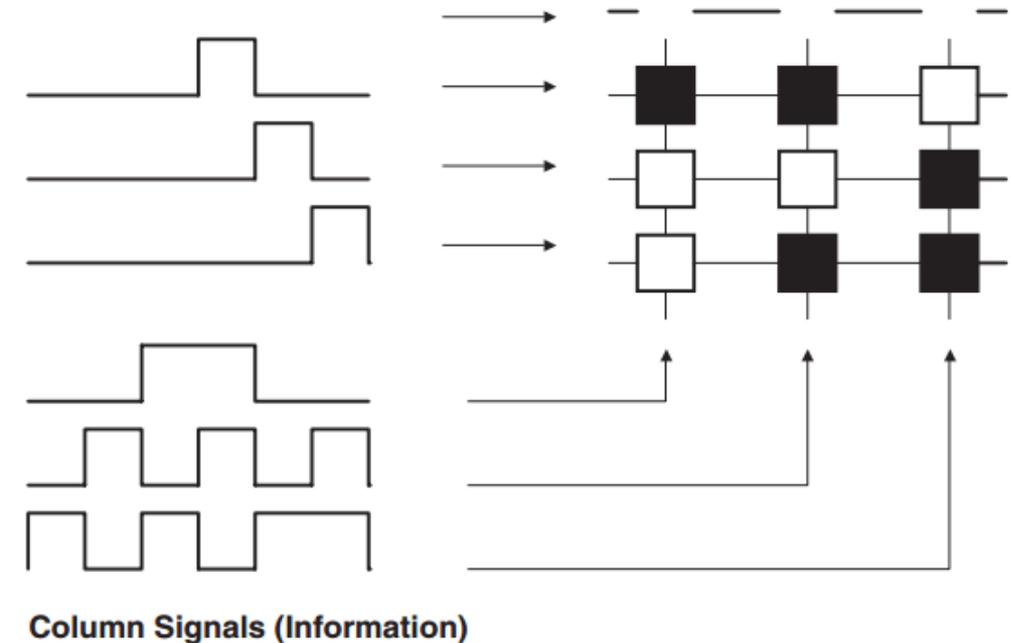
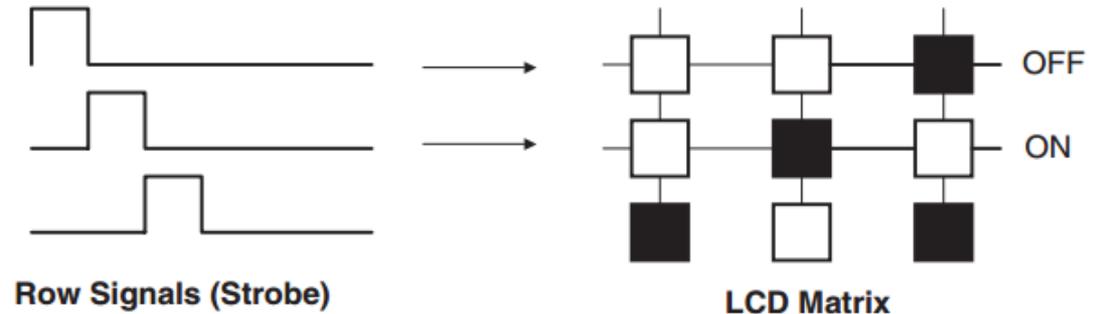


Vantagens/Desvantagens do LCD

- Vantagens:
 - Possuem uma tela plana, eliminando as distorções de imagem encontrados em monitores do tipo tubo de raios catódicos
 - Cansam menos a vista
 - Consomem menos energia
 - Emitem pouquíssima radiação nociva
 - São baratos
- Desvantagens:
 - Persistência pode levar a efeitos de "arrasto" na exibição de imagens com movimento
 - Visualização com ângulo limitado
 - Pixels mortos ou presos podem ocorrer durante a fabricação
 - Perda de brilho e tempo de resposta lento em ambientes com baixa temperatura
 - Perda de contraste em ambientes com alta temperatura.
 - O visor não funciona corretamente na luz solar direta

LCD de Matriz Passiva

- Elementos da imagem (pixels) são separados em linhas e colunas
- Linhas são varridas sequencialmente
- Informação é colocada no sinal das colunas
- Elementos ficam “acesos” apenas uma pequena parcela de tempo, causando perda de brilho e contraste
- Podem empregar **endereçamento direto**, quando cada pixel tem alimentação separada



LCD de Matriz Ativa

- Possui um transistor para cada pixel da imagem
- Transistores são fabricados na técnica TFT (*Thin Film Transistor*)
- Sinal que é enviado ao pixel é mantido pelo transistor até o próximo *refresh*
- Melhora significativamente o brilho e contraste da imagem
- Tela VGA precisa de 921.000 transistores (640 x 480 x 3)
- Tela XGA precisa de 2.359.296 transistores (1024 x 768 x 3)
- A matriz completa de transistores precisa ser fabricada em uma única peça
- Presença de imperfeições fazem com que exista uma alta taxa de descarte
- Por este motivo telas TFT são caras

LCD Embarcado

- Módulos de LCD são muito utilizados em sistemas embarcados
 - Incluem a tela e os circuitos necessários ao seu funcionamento
- Módulos podem ser **gráficos** e **alfanuméricos** (caracteres)
- Módulos gráficos:
 - Resoluções: 122 x 32, 128 x 64, 240 x 64, 240 x 128 pixels
 - Interface com conector de 20 pinos
- Módulos alfanuméricos:
 - Número de colunas x linhas: 8 x 2, 12 x 2, 16 x 1, 16 x 2, 16 x 4, 20 x 1, 20 x 2, 20 x 4, 24 x 2, 24 x 4, 40 x 2, 40 x 4
 - Interface com conector de 14 a 16 pinos
- **Neste curso:** apenas módulos alfanuméricos

HD44780U

- Controlador/Driver para Dot Matrix LCD alfanumérico
- Mostra letras, números, símbolos e caracteres kana (língua japonesa)
- Interface com 4 ou 8 bits com velocidade de até 2 MHz
- Baixo consumo
- Diversas funções para facilitar programação e controle
- Possui 80 bytes de RAM de dados (*Display Data* RAM, DDRAM), 64 bytes de RAM e 1240 bytes de ROM para o Gerador de Caracteres (*Character Generator* RAM e ROM, CGRAM e CGROM)

HD44780U

- Entrada *Register Select* (**RS**): seleciona acesso ao registrador de instrução (RS = 0) ou dados (RS = 1)
- Entrada *Read/Write* (**R/W**): seleciona acesso de leitura (R/W = 1) ou escrita (R/W = 0)
- Entrada/Saída *Data Bus* (**DB4:7**): 4 bits mais significativos do barramento de dados
- Entrada/Saída *Data Bus* (**DB0:3**): 4 bits menos significativos do barramento de dados; não são usados para interface de 4 bits
- Entrada *Enable* (**E**): habilita o acesso aos registradores

HD44780U

- Possui dois registradores: Instrução (**IR**) e Dado (**DR**)
 - IR é para escrita de comandos, como limpar display, mudar o cursor, etc.
 - IR é apenas para escrita
 - DR armazena temporariamente dados escritos e lidos da memória do módulo
- Leitura de dados
 - Quando informação de endereço é escrito no IR, o dado correspondente é colocado no DR
 - Após a leitura do DR pelo processador, o próximo endereço é colocado no DR
- Bit *Busy Flag* (**BF**)
 - Quando $BF = 1$, o HD44780U está ocupado em operações internas e não aceita novas instruções
 - Para uma instrução ser aceita, BF deve ser 0

HD44780U

- *Address Counter (AC)*
 - Usado para acessar DDRAM e CGRAM
 - Conteúdo de AC é incrementado em 1 após acesso

RS	R/W	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

- *Character Generator ROM (CGROM)*
 - Onde estão definidos os formatos das letras, números e símbolos em padrões 5 x 8 e 5 x 10 pontos
- *Character Generator RAM (CGRAM)*
 - Permite ao usuário redefinir até 8 caracteres 5 x 8 pontos ou até 4 caracteres 5 x 10 pontos

HD44780U

- *Data Display* RAM (**DDRAM**)
 - Contém os caracteres (até 80) mostrados pelo display
 - Para display de 8 caracteres x 1 linha:

Display position (digit)	1	2	3	4	5	79	80
DDRAM address (hexadecimal)	00	01	02	03	04	4E	4F

Display position	1	2	3	4	5	6	7	8
DDRAM address	00	01	02	03	04	05	06	07

For shift left	01	02	03	04	05	06	07	08
-------------------	----	----	----	----	----	----	----	----

For shift right	4F	00	01	02	03	04	05	06
--------------------	----	----	----	----	----	----	----	----

HD44780U

- *Data Display* RAM (**DDRAM**)
 - Para display de 8 caracteres x 2 linhas:

Display position	1	2	3	4	5			39	40
DDRAM address (hexadecimal)	00	01	02	03	04		26	27
	40	41	42	43	44		66	67

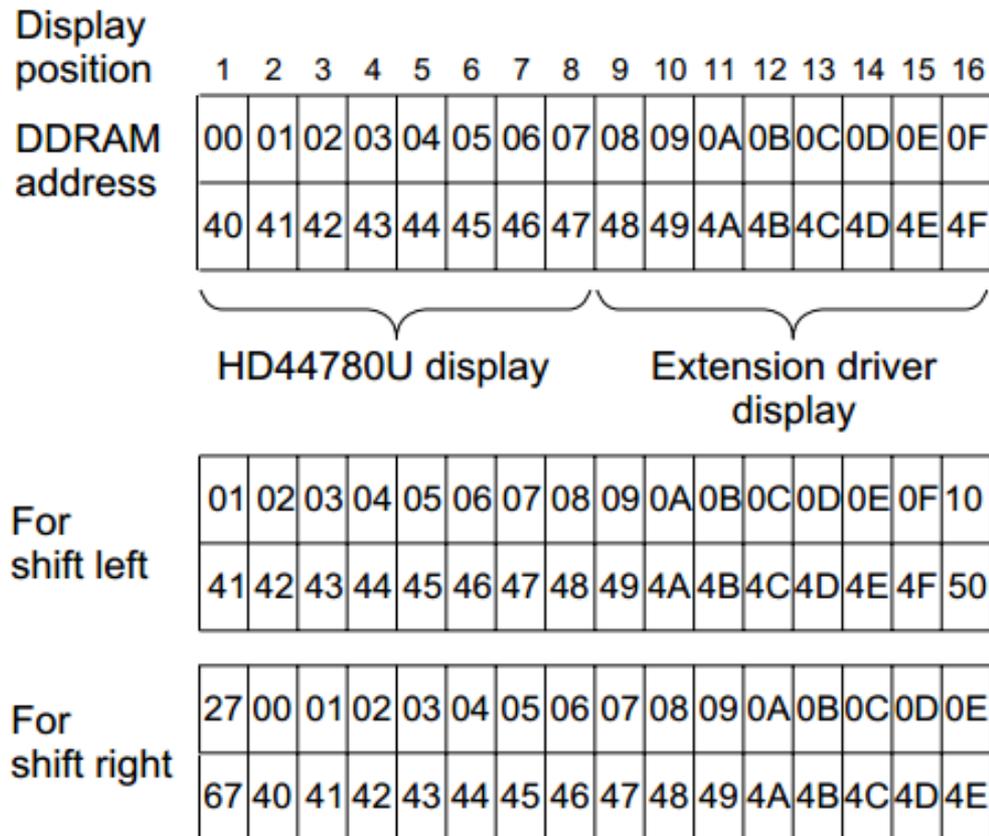
Display position	1	2	3	4	5	6	7	8
DDRAM address	00	01	02	03	04	05	06	07
	40	41	42	43	44	45	46	47

For shift left	01	02	03	04	05	06	07	08
	41	42	43	44	45	46	47	48

For shift right	27	00	01	02	03	04	05	06
	67	40	41	42	43	44	45	46

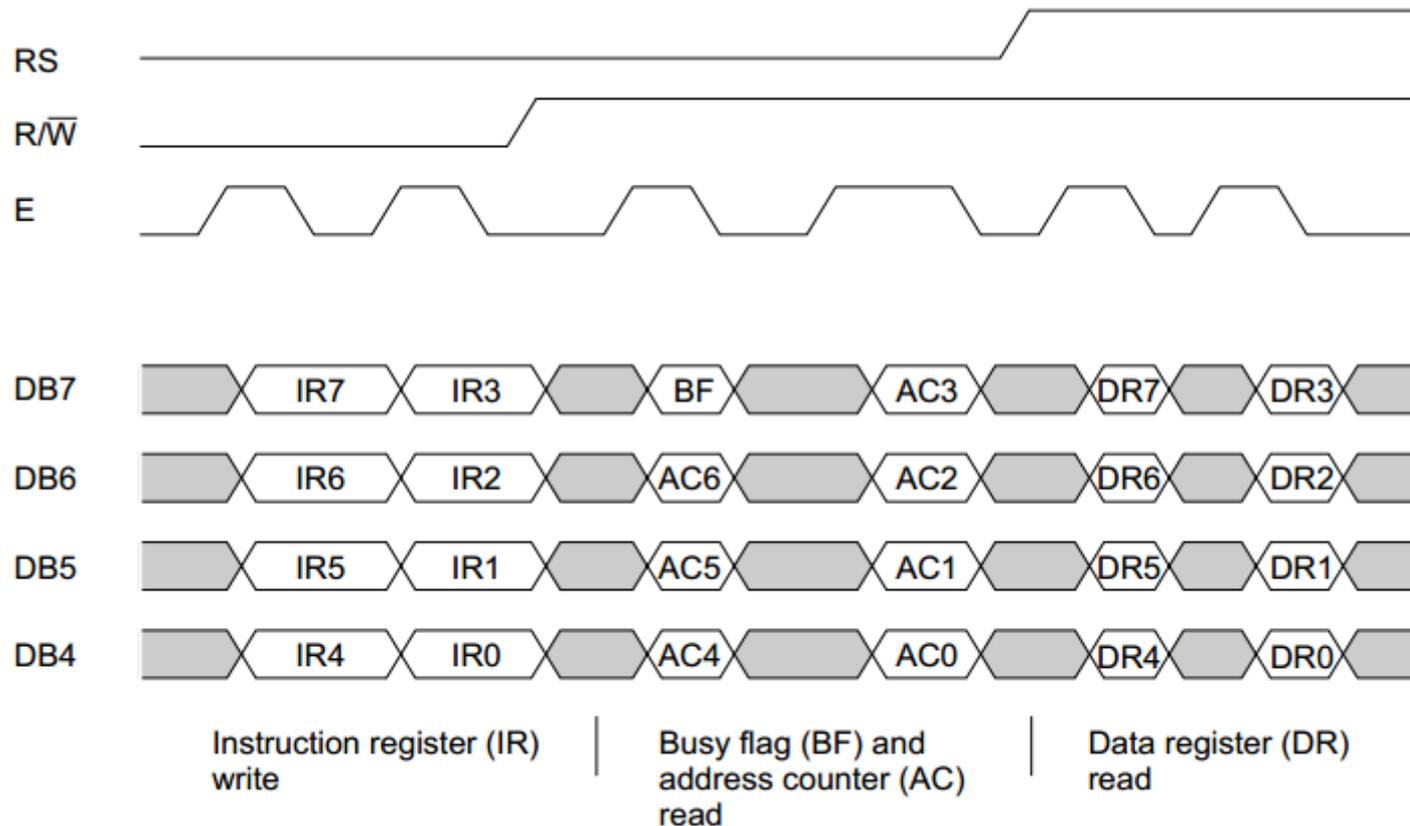
HD44780U

- *Data Display* RAM (**DDRAM**)
 - Para display de 16 caracteres x 2 linhas:



HD44780U

- Cursor aparece no caracter da DDRAM apontado pelo AC
- Interface com o processador:



Inicialização do HD44780U

- Rotina executada ao ligar o display:
 - Executa Limpa display
 - Configura:
 - DL = 1 (interface de 8-bit)
 - N = 0 (display com 1 linha)
 - F = 0 (caracter 5 x 8)
 - D = 0 (display desligado)
 - C = 0 (cursor desligado)
 - B = 0 (sem piscas)
 - I/D = 1 (incremento por 1)
 - S = 0 (sem deslocamento)
- Tempo de inicialização: 10 ms
 - durante este tempo BF = 1

Instruções do HD44780U

Instruction	Code										Description	Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	1.52 ms
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 μ s
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 μ s
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 μ s

Instruções do HD44780U

Instruction	Code										Description	Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 μ s
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 μ s
Set DDRAM address	0	0	1	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 μ s						
Read busy flag & address	0	1	BF	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 μ s						
Write data to CG or DDRAM	1	0	Write data								Writes data into DDRAM or CGRAM.	37 μ s $t_{ADD} = 4 \mu$ s*
Read data from CG or DDRAM	1	1	Read data								Reads data from DDRAM or CGRAM.	37 μ s $t_{ADD} = 4 \mu$ s*

Instruções do HD44780U

I/D = 1: Increment
I/D = 0: Decrement
S = 1: Accompanies display shift
S/C = 1: Display shift
S/C = 0: Cursor move
R/L = 1: Shift to the right
R/L = 0: Shift to the left
DL = 1: 8 bits, DL = 0: 4 bits
N = 1: 2 lines, N = 0: 1 line
F = 1: 5 × 10 dots, F = 0: 5 × 8 dots
BF = 1: Internally operating
BF = 0: Instructions acceptable

DDRAM: Display data RAM
CGRAM: Character generator RAM
ACG: CGRAM address
ADD: DDRAM address
(corresponds to cursor address)
AC: Address counter used for both DD and CGRAM addresses

Execution time changes when frequency changes
Example:
When f_{cp} or f_{osc} is 250 kHz,
 $37 \mu s \times \frac{270}{250} = 40 \mu s$

Módulo LCD com HD44780U

- Interface com os sinais **RS** (*Register Select*), **R/W** (*Read/Write*), **DB0:7** (*Data Bus*) e **E** (*Enable*)
- Entrada **LED+** (anodo) e **LED-** (catodo): LED de iluminação de fundo (*back light*)
- Entrada **V0**: controle de contraste
- Entrada **VSS** (terra) e **VDD** (5 V): alimentação

Módulo HD44780U com Arduino

- Ligações do módulo com o Arduino para interface de 4 bits

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

Biblioteca Arduino LCD

- **LiquidCrystal()**

- Cria uma variável do tipo LiquidCrystal
- Na sintaxe, omite os parâmetros que não são usados
- 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)
 - Parâmetros:
 - rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7 deve conter o pino do Arduino conectado ao sinal

Biblioteca LiquidCrystal

- **lcd.begin()**
 - Inicializa a interface com o LCD e especifica as dimensões
 - Deve ser chamada antes de qualquer outro comando da biblioteca LCD
 - Sintaxe: `lcd.begin(cols, rows)`
 - Parâmetros:
 - lcd: uma variável de tipo `LiquidCrystal`
 - cols: número de colunas
 - rows: número de linhas

Biblioteca LiquidCrystal

- **lcd.clear()**
 - Limpa a tela do LCD e coloca o cursor no canto superior esquerdo
 - Sintaxe: `lcd.clear()`
- **lcd.home()**
 - Coloca o cursor no canto superior esquerdo
 - Sintaxe: `lcd.home()`
- **lcd.write()**
 - Escreve um caracter no display na posição do cursor
 - Sintaxe: `lcd.write(data)`
 - Parametros:
 - data: caractere a ser escrito

Biblioteca LiquidCrystal

- **lcd.setCursor()**

- Coloca cursor na posição indicada
- Sintaxe: `lcd.setCursor(col, row)`
 - Parametros:
 - col: número da coluna (primeira coluna é 0)
 - row: número da linha (primeira linha é 0)

- **lcd.print()**

- Imprime texto no display
- Sintaxe:
 - `lcd.print(data)`
 - `lcd.print(data, BASE)`
 - Parametros:
 - data: dado a ser impresso (char, byte, int, long ou string)
 - BASE (opcional): a base dos números: BIN para binário, DEC para decimal, OCT para octal, HEX para hexadecimal

Biblioteca LiquidCrystal

- **lcd.cursor()** e **lcd.noCursor()**
 - Mostra ou omite o cursor
 - Sintaxe: `lcd.cursor()` e `lcd.noCursor()`
- **lcd.blink()** e **lcd.noBlink()**
 - Mostra um cursor piscante ou não piscante
 - Sintaxe: `lcd.blink()` e `lcd.noBlink()`
- **lcd.display()** e **lcd.noDisplay()**
 - Liga e desliga o display, mantendo o texto e posição do cursor
 - Sintaxe: `lcd.display()` e `lcd.noDisplay()`

Biblioteca LiquidCrystal

- **lcd.scrollDisplayLeft()** e **lcd.scrollDisplayRight()**
 - Rola o conteúdo do display uma posição para a esquerda e direita
 - Sintaxe: `lcd.scrollDisplayLeft()` e `lcd.scrollDisplayRight()`
- **lcd.autoscroll()** e **lcd.noAutoscroll()**
 - Liga ou desliga a rolagem automática do texto
 - Escrever no display com o `autoscroll` ligado faz com que o texto role de maneira que o último caractere escrito fique visível
 - Desligar o `autoscroll` significa que o display não rola
 - Sintaxe: `lcd.autoscroll()` e `lcd.noAutoscroll()`

Biblioteca LiquidCrystal

- **lcd.leftToRight()** e **lcd.rightToLeft()**
 - Determina a direção como o texto é escrito no display
 - Sintaxe: `lcd.leftToRight()` e `lcd.rightToLeft()`
- **lcd.createChar()**
 - Cria um novo caractere para uso no display
 - Até 8 caracteres (numerados de 0 a 7) de 5 x 8 pontos podem ser criados
 - Caractere é especificado por um array de 8 bytes, um por linha
 - Sintaxe: `lcd.createChar(num, data)`
 - Parametros:
 - num: número do caracter a ser criado
 - data: dado do caracter