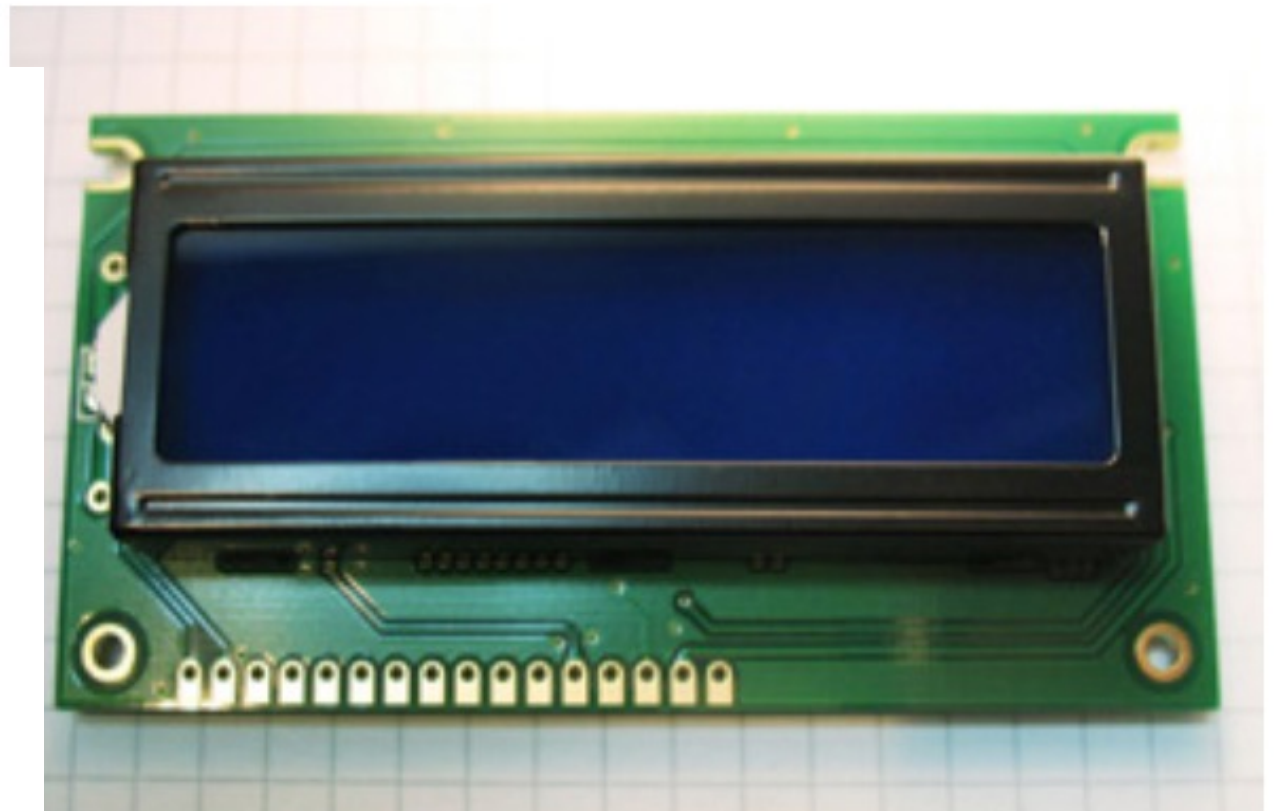


AVR Microcontrolador + LCD 16x2

Prof. Marcos Chaves

DISPLAY DE CRISTAL LÍQUIDO ALFANUMÉRICO



- Interface comum em sistemas microprocessador
- Apresenta uma série de configurações de linhas e colunas
 - 16x1, 16x2, 20x2, 20x4, 8x2



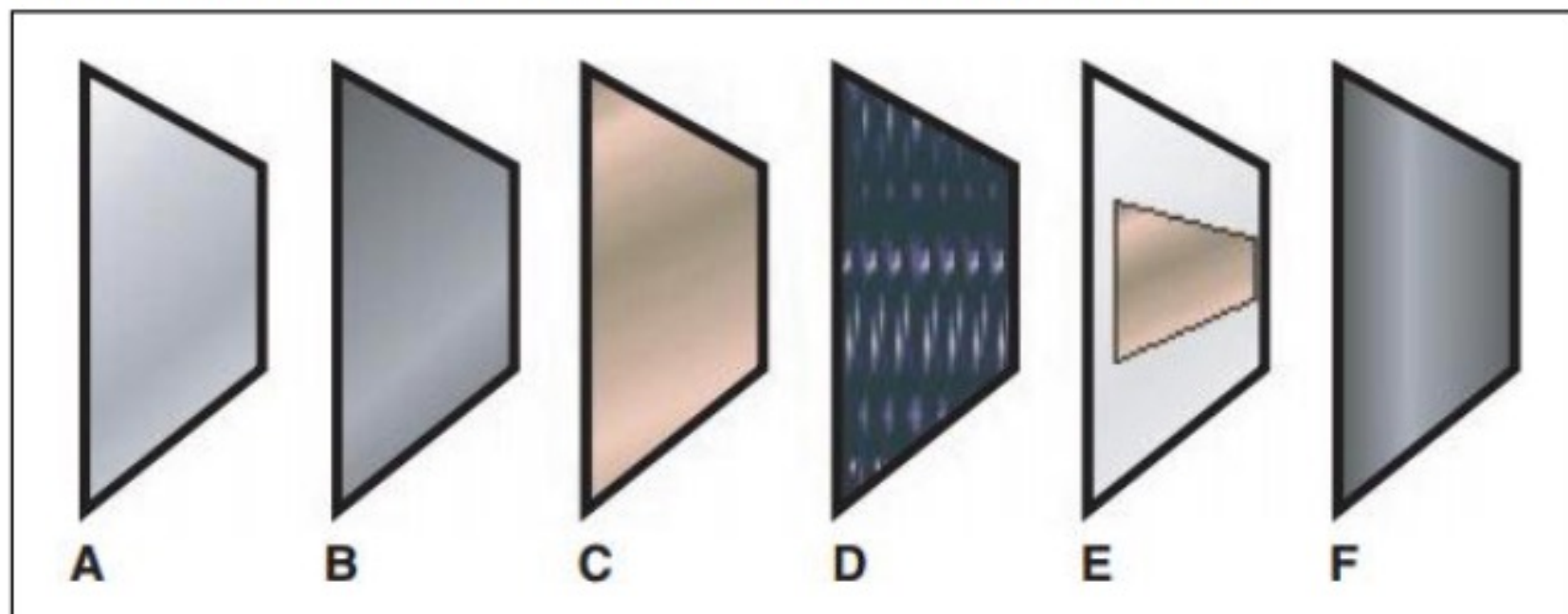
Histórico do LCD

- 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 Heilmeier 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



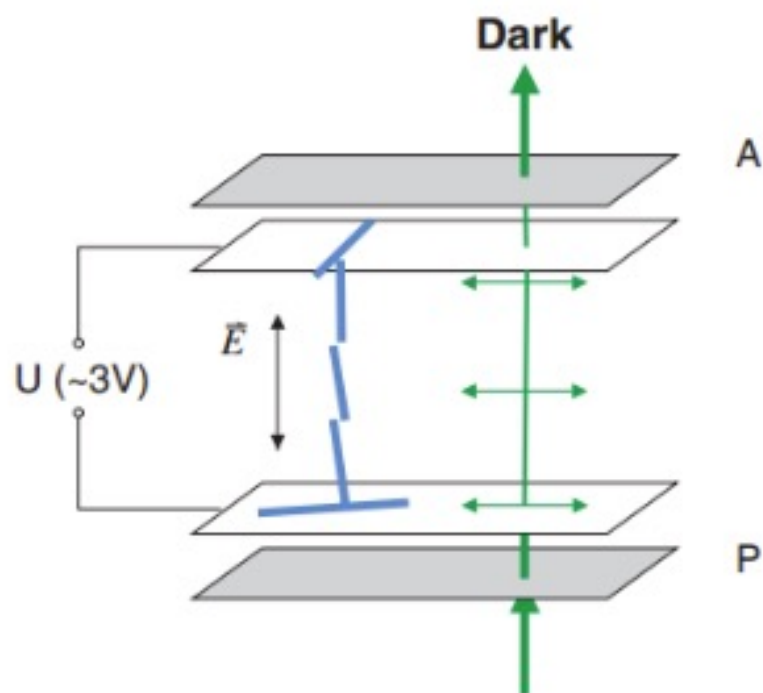
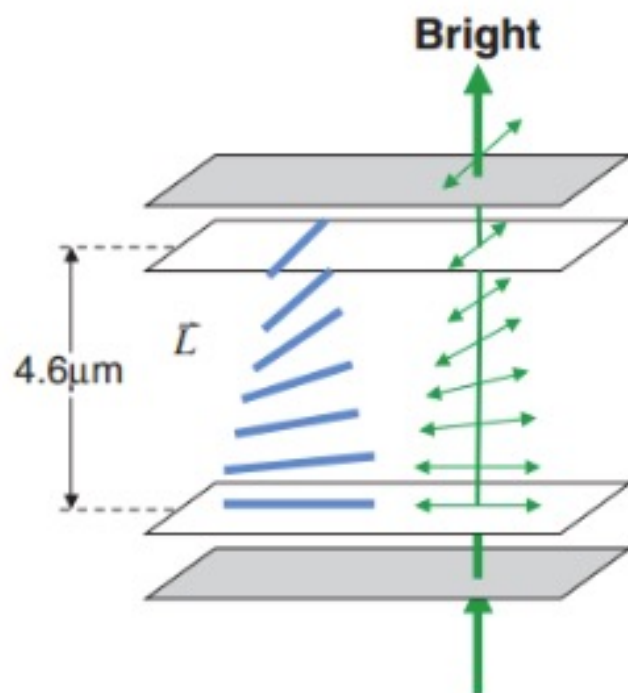
Como é construído?

- 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?

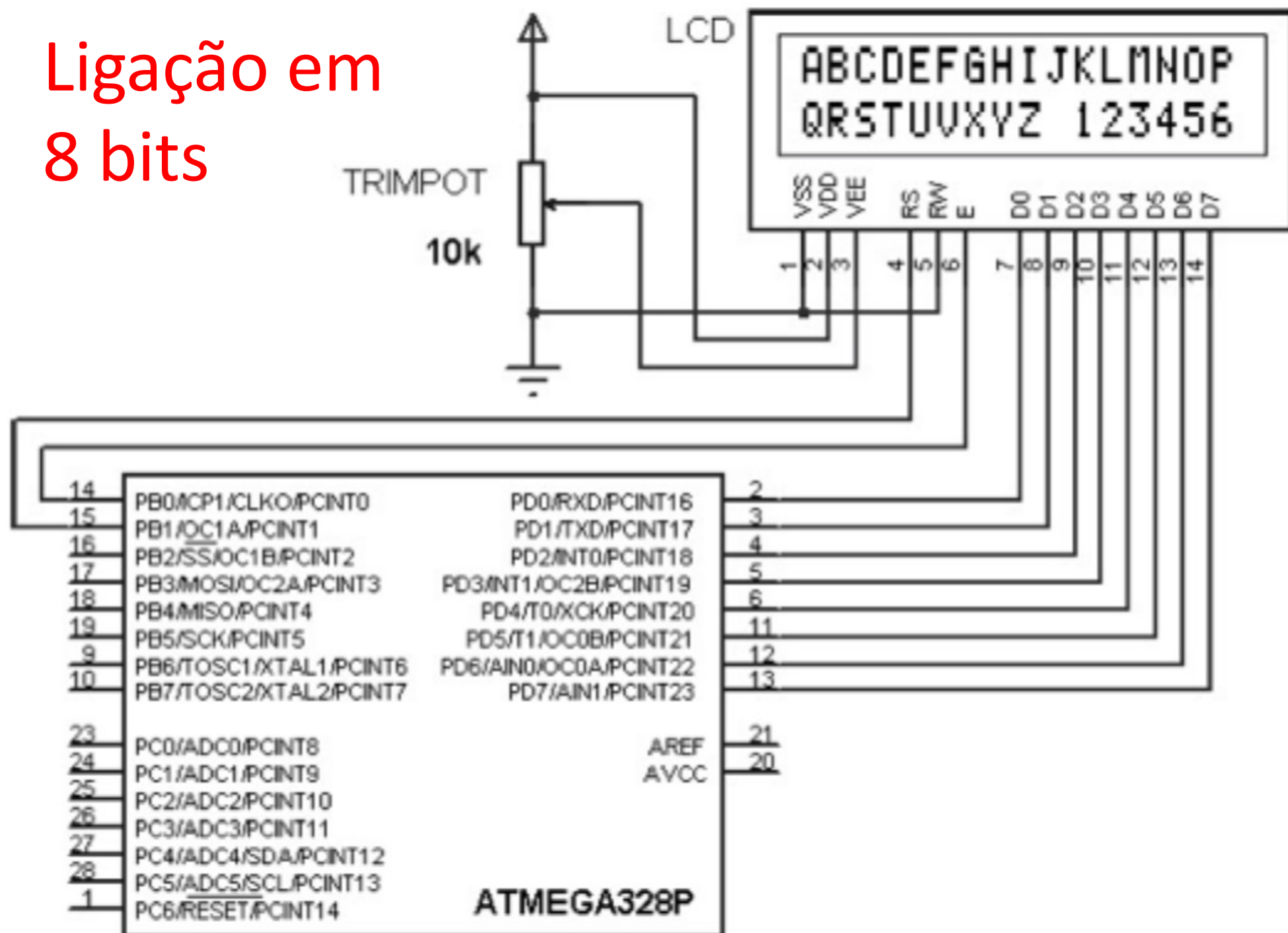
- Aplicação de tensão no eletrodo (E) permite ou impede a passagem de luz
- Cristais líquidos não emitem luz
- **LCD reflexivo**: mais baratos, possui espelho no fundo e apenas reflete a luz externa
 - Funcionam melhor em ambiente bem iluminado
- **LCD back lit**: possui iluminação colocada acima, ao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

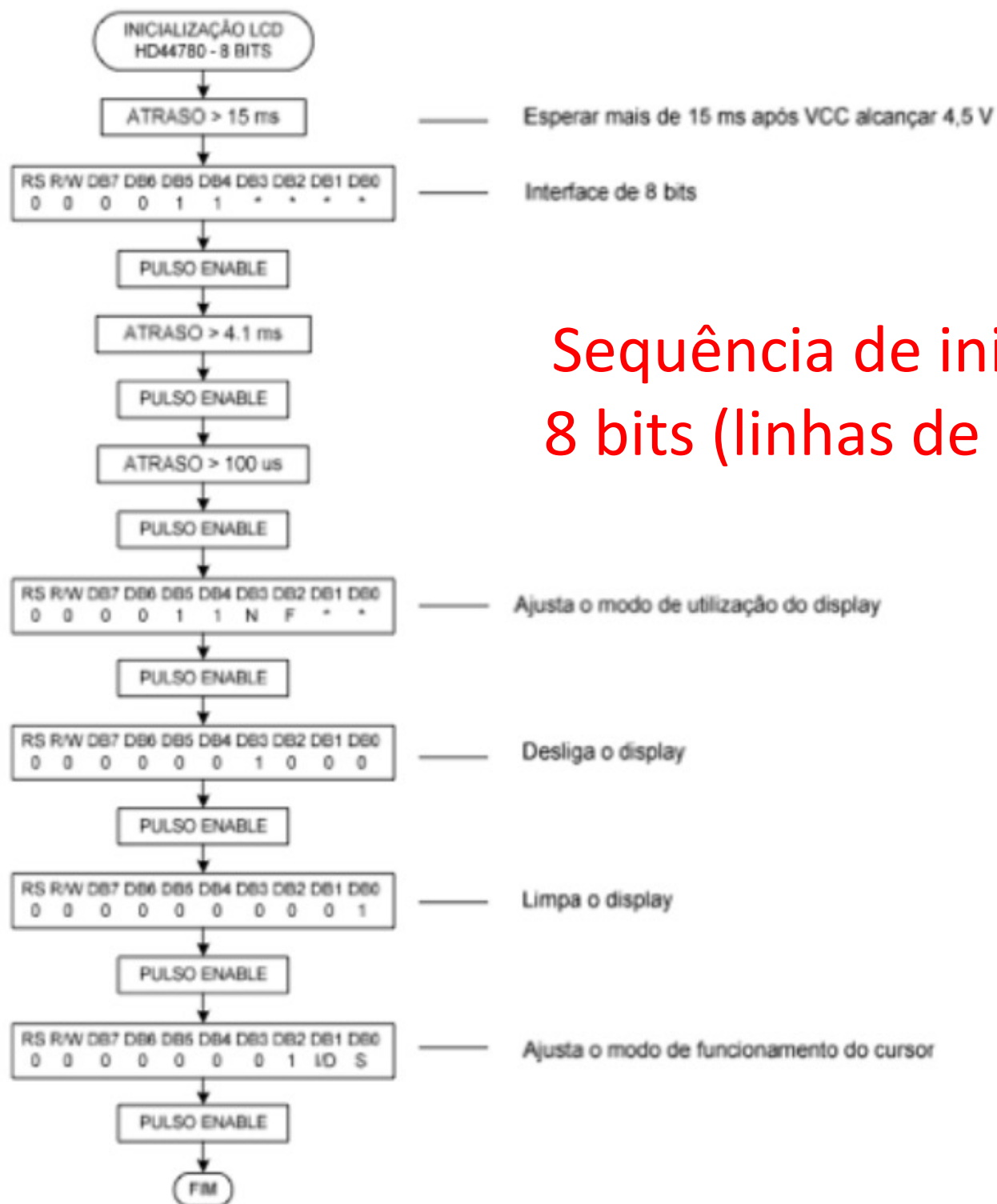




- Utiliza uma interface paralela de 8bits para leitura e escrita no LCD
 - Controlador HD44780 (Hitachi)
 - Pode ser configurado para utilizar apenas 4bits [economia de pinos de I/O]
- Possui controle de contraste do display de cristal líquido
- Alguns módulos podem apresentar um led de retro-iluminação (backlight)

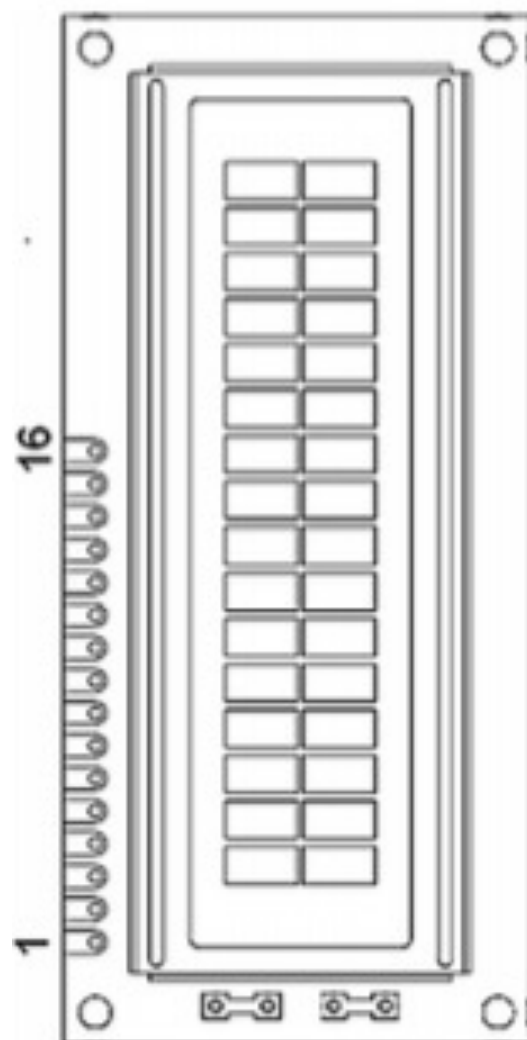
Ligação em
8 bits





Sequência de inicialização em 8 bits (linhas de comunicação)

Tab. B1: Pinagem de um LCD 16 × 2.



Pino	Função	Descrição
1	Alimentação	VSS (GND)
2	Alimentação	VCC
3	VEE	Tensão para ajuste do contraste do LCD
4	RS	<i>Register Select</i> : 1 = dado, 0 = instrução
5	R/W	<i>Read/Write</i> : 1 = leitura, 0 = escrita
6	E	<i>Enable</i> : 1 = habilita, 0 = desabilita
7	DB0	Barramento de dados
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	LED+ (A)	Anodo do LED de iluminação de fundo
16	LED - (K)	Catodo do LED de iluminação de fundo

- Interface pode ler/escrever comandos ou dados
- Comandos
 - Configuração do modo de operação
 - Manipulação do Cursos
 - Deslocamento da posição
 - Limpeza do Display

RS=0
comando

Tab. B3: Resumo dos códigos de instruções.

Descrição	Modo	Código Hexa
Controle do display	Liga (sem cursor)	0x0C
	Desliga	0x0A/0x08
Limpa display com retorno do cursor		0x01
Controle do cursor	Liga	0x0E
	Desliga	0x0C
	Desloca p/ a esquerda	0x10
	Desloca p/ a direita	0x14
	Retorno	0x02
	Cursor piscante	0x0D
	Cursor com alternância	0x0F
Sentido de deslocamento do cursor na entrada de um caractere	Para a esquerda	0x04
	Para a direita	0x06
Deslocamento da mensagem na entrada de um caractere	Para a esquerda	0x07
	Para a direita	0x05
Deslocamento da mensagem sem a entrada de caractere	Para a esquerda	0x18
	Para a direita	0x1C
Endereço da primeira posição do cursor	Primeira linha	0x80
	Segunda linha	0xC0

RS=0
comando

Posição do cursor

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Linha 1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
Linha 2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

Fig. 5.21 – Endereços para escrita num LCD 16 × 2.

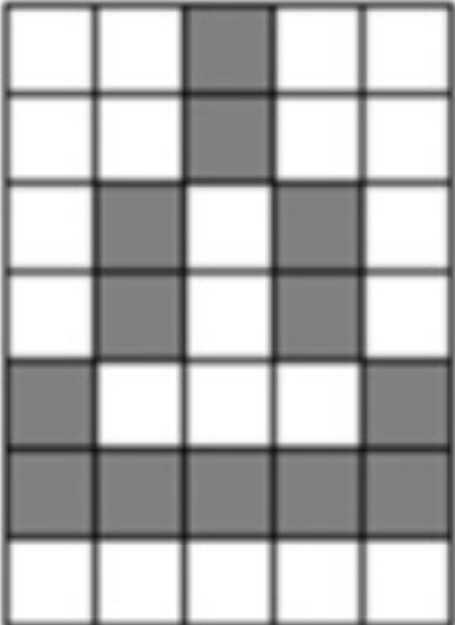
RS=1
carattere

NIBBLE ALTO NIBBLE BAIXO		0_	1_	2_	3_	4_	5_	6_	7_	8_	9_	A_	B_	C_	D_	E_	F_		
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111		
_0	xxxx 0000	CG RAM (1)			0	0	B	P	z	F				-	ダ	エ	0	P	
_1	xxxx 0001	CG RAM (2)			!	1	A	Q	a	q				ア	チ	4	3	q	
_2	xxxx 0010	CG RAM (3)			"	2	B	R	b	r				イ	ツ	x	0	0	
_3	xxxx 0011	CG RAM (4)			#	3	C	S	c	s				ウ	テ	E	e	x	
_4	xxxx 0100	CG RAM (5)			\$	4	D	T	d	t				エ	ト	h	w	0	
_5	xxxx 0101	CG RAM (6)			%	5	E	U	e	u				オ	タ	1	0	0	
_6	xxxx 0110	CG RAM (7)			&	6	F	V	f	v				ヲ	カ	ニ	3	0	Z
_7	xxxx 0111	CG RAM (8)			'	7	G	W	g	w				ア	チ	ヌ	ラ	q	π
_8	xxxx 1000	(1)			(8	H	X	h	x				イ	ウ	ホ	リ	フ	X
_9	xxxx 1001	(2))	9	I	Y	i	y				ウ	ケ	ル	リ	フ	
_A	xxxx 1010	(3)			*	:	J	Z	j	z				エ	コ	ハ	V	i	フ
_B	xxxx 1011	(4)			+	:	K	E	k	e				オ	サ	ヒ	コ	x	π
_C	xxxx 1100	(5)			,	<	L	x	l					ホ	ロ	フ	フ	x	π
_D	xxxx 1101	(6)			-	=	M	I	m					ユ	ズ	ハ	フ	キ	÷
_E	xxxx 1110	(7)			.	>	N	^	n	+				ヨ	セ	ホ	キ	π	
_F	xxxx 1111	(8)			/	?	O	_	o	+				ウ	リ	マ	マ	0	■

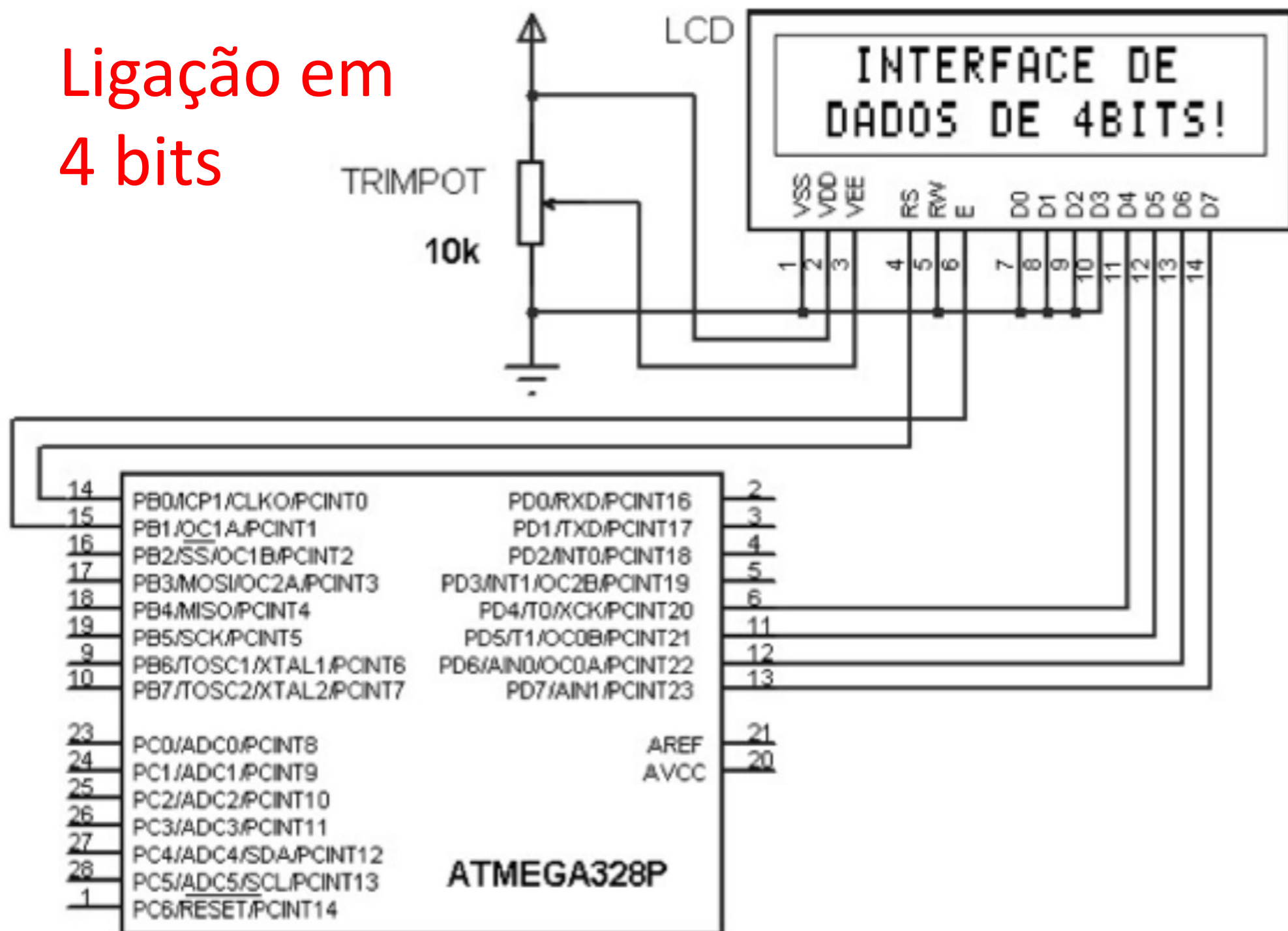
Lower 4 Bits	Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	@	P	`	F	B	α		°	À	Ð	à	ä	
	(2)		!	1	A	Q	a	q	A	J	i	±	Á	Ñ	á	ñ	
xxxx0010	(3)	“	"	2	B	R	b	r	Ж	Г	¢	²	Â	Ò	â	ô	
	(4)	”	#	3	C	S	c	s	З	π	£	³	Ã	Ó	ã	õ	
xxxx0100	(5)		\$	4	D	T	d	t	И	Σ	¤	₣	Ä	Ö	ä	ö	
	(6)		%	5	E	U	e	u	Й	σ	¥	₤	Å	Õ	å	ö	
xxxx0110	(7)		&	6	F	V	f	v	Ј	Ј	!	¶	Æ	Ö	æ	ö	
	(8)		'	7	G	W	g	w	П	τ	§	•	Ç	×	ç	÷	
xxxx1000	(1)	↑	(8	H	X	h	x	У	‡	†	ω	É	Æ	é	ø	
	(2)	↓)	9	I	Y	i	y	Ч	Θ	‡	¹	Ê	Ü	ê	ü	
xxxx1010	(3)	→	*	:	J	Z	j	z	Ч	Ω	≡	Ω	Ê	Ú	ê	ú	
	(4)	←	+	:	K	[k	{	Ш	δ	※	※	Ë	Û	ë	û	
xxxx1100	(5)	≤	,	<	L	\	l		Щ	≈	№	¼	İ	Ü	ı	ü	
	(6)	≥	-	=	M]m	}	b	‡	Я	½	ı	Ý	ı	ı	ı	
xxxx1110	(7)		.	>	N	^	n	~	Ы	ε	¼	ı	İ	İ	İ	İ	
	(8)		/	?	O	_	o	ó	Ω	Π	'	¿	İ	İ	İ	İ	

Lower 4 Bits \ Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	a	P	`	P				-	タ	ミ	α	ρ
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	△	ä	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	×	β	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	ω
xxxx0100	(5)		\$	4	D	T	d	t			、	エ	ト	ホ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	1	℃	Ü
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			フ	キ	ヌ	ラ	g	π
xxxx1000	(1)		(8	H	X	h	x			ィ	ク	ネ	リ	フ	×
xxxx1001	(2))	9	I	Y	i	y			ウ	ケ	リ	ル	´	4
xxxx1010	(3)		*	:	J	Z	j	z			エ	コ	ハ	レ	j	チ
xxxx1011	(4)		+	;	K	L	k	l			オ	サ	ヒ	ロ	*	斤
xxxx1100	(5)		,	<	L	¥	1	l			ヤ	シ	フ	ワ	Φ	円
xxxx1101	(6)		-	=	M	I	m	}			ユ	ス	ハ	ン	も	÷
xxxx1110	(7)		.	>	N	^	n	÷			ヨ	セ	ホ	°	ñ	
xxxx1111	(8)		/	?	O	_	o	+			ッ	ソ	マ	°	ö	■

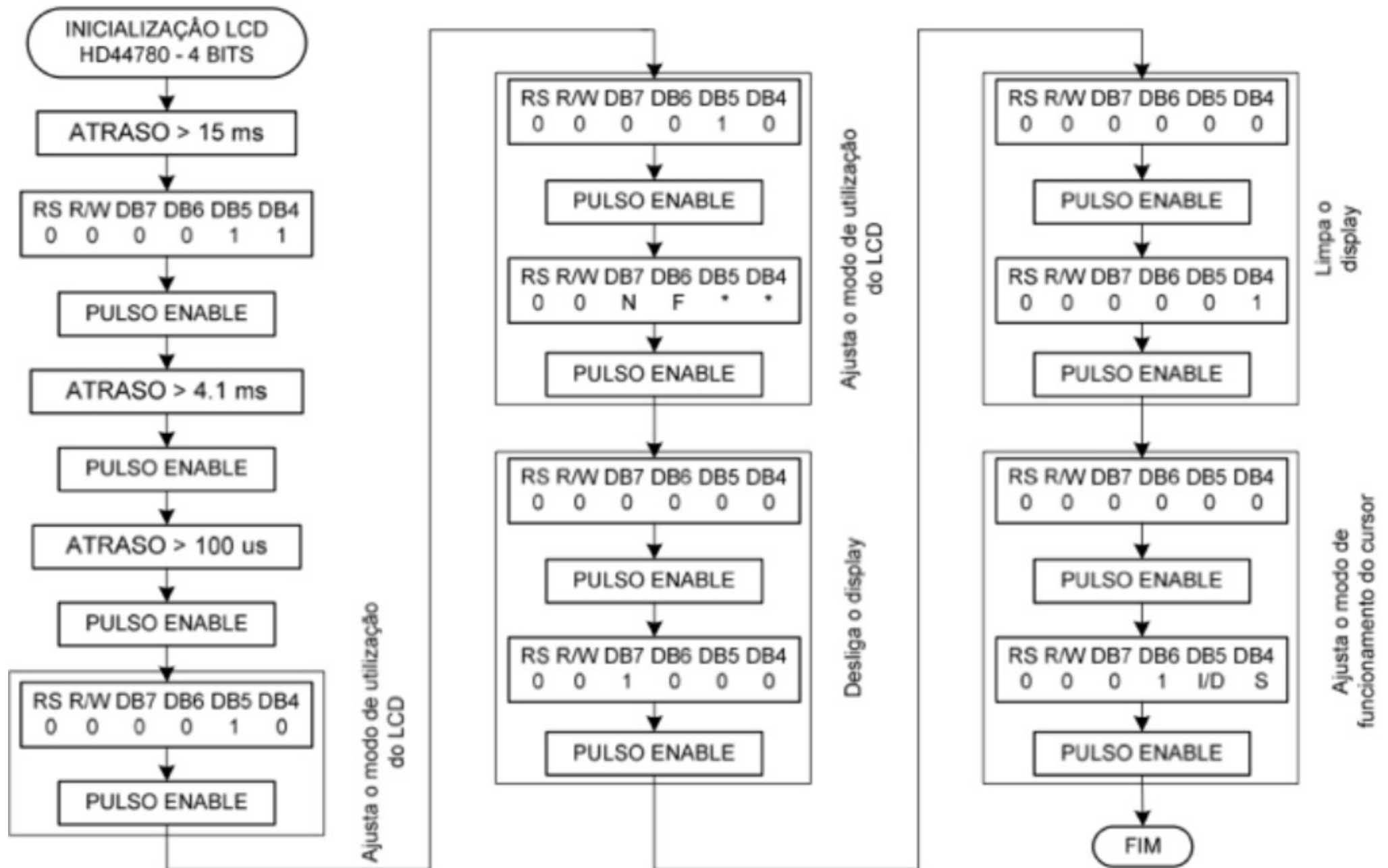
Caracter do usuário (Ram LCD)

Endereço da CGRAM	Mapa de bits	Dado
0x48		0b00100
0x49		0b00100
0x4A		0b01010
0x4B		0b01010
0x4C		0b10001
0x4E		0b11111
0x4F		0b00000

Ligação em 4 bits



Sequência de inicialização em 4 bits



Como Utilizar a biblioteca para LCD em assembly

; incluir biblioteca

.include "lib328Pv03.inc"

; inicializacao LCD em 4 bits

rcall lcd_init

; Chama rotina limpar o LCD e posicionar na linha 0, coluna 0

rcall lcd_clear

Como Utilizar a biblioteca para LCD em assembly

;;;;;;;;; posiciona cursor

ldi lcd_col,3 ;define coluna3

rcall lcd_lin0_col ;define linha 0

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Linha 1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
Linha 2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

Fig. 5.21 – Endereços para escrita num LCD 16 × 2.

Utilizando a biblioteca para LCD em assembly

;;;;;;;;;;;;; escreve mensagem letra por letra

ldi lcd_character,'I' ;; carrega letra entre aspas

rcall lcd_write_character. ; chama rotina para imprimir caracter

ldi lcd_character,'F'

rcall lcd_write_character

ldi lcd_character,'S'

rcall lcd_write_character

ldi lcd_character,'P'

rcall lcd_write_character

;;;;;;;;;;;;; imprimir numero

ldi r16,10 ; carregue o numero em um registrador

mov lcd_number, r16 ;;; move para o registro da biblioteca LCD

rcall lcd_write_number ;; chama rotina para imprimir numero