

# EEL7030 - Microprocessadores



**Prof. Raimes Moraes**

**GpqCom – EEL**

**UFSC**

# **Acesso a dispositivos de entrada e saída**

**As portas de um microcontrolador são uteis para escrever em leds, ler de chaves, como também interagir com periféricos mais complexos, tais como display de cristal líquido (LCD)**

## Endereço da Área Visível de LCDs de 2 linhas



Geralmente, possui periférico para escrever pixel por pixel de forma a compor o caractere.

Display	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16						
Line 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	...
Line 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	...

8051  
Micro

P1.0 DB0  
P1.1 DB1  
P1.2 DB2  
P1.3 DB3  
P1.4 DB4  
P1.5 DB5  
P1.6 DB6  
P1.7 DB7

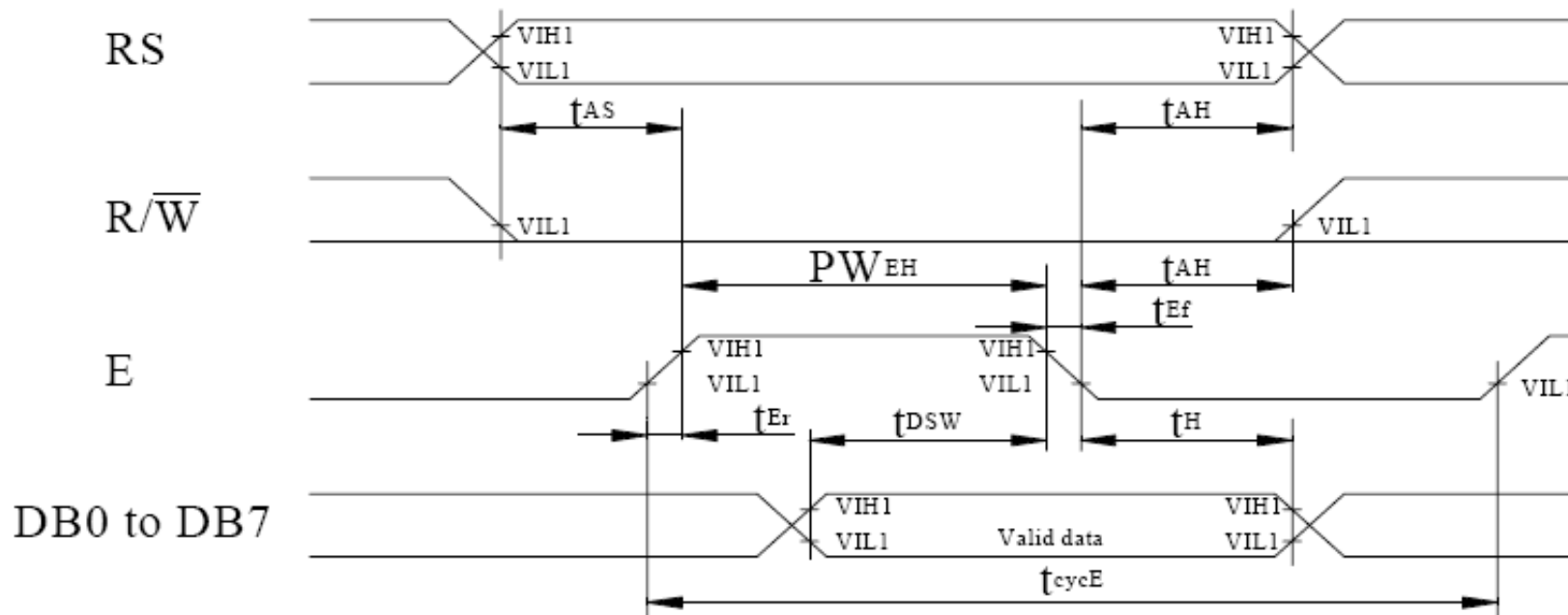
HD44780  
LCD

P3.7 EN  
P3.6 RS  
P3.5 RW

Conexão e procedimento de  
escrita nos registradores internos  
do HD44780 que controla o LCD

--  
Interface de 8 pinos

RS= '0' : comando;  
RS= '1' : dado;



Escreva 4 bits MSB

Escreva 4 bits LSB

**DDRAM = Display Data RAM**  
**CGRAM Character Generator RAM**

P1  
.3

P1  
.7

P1  
.6

P1  
.5

P1  
.4

P1  
.7

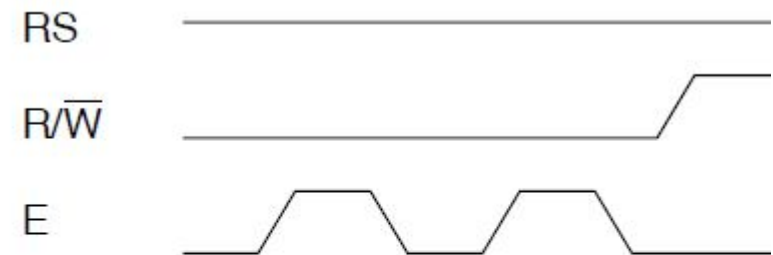
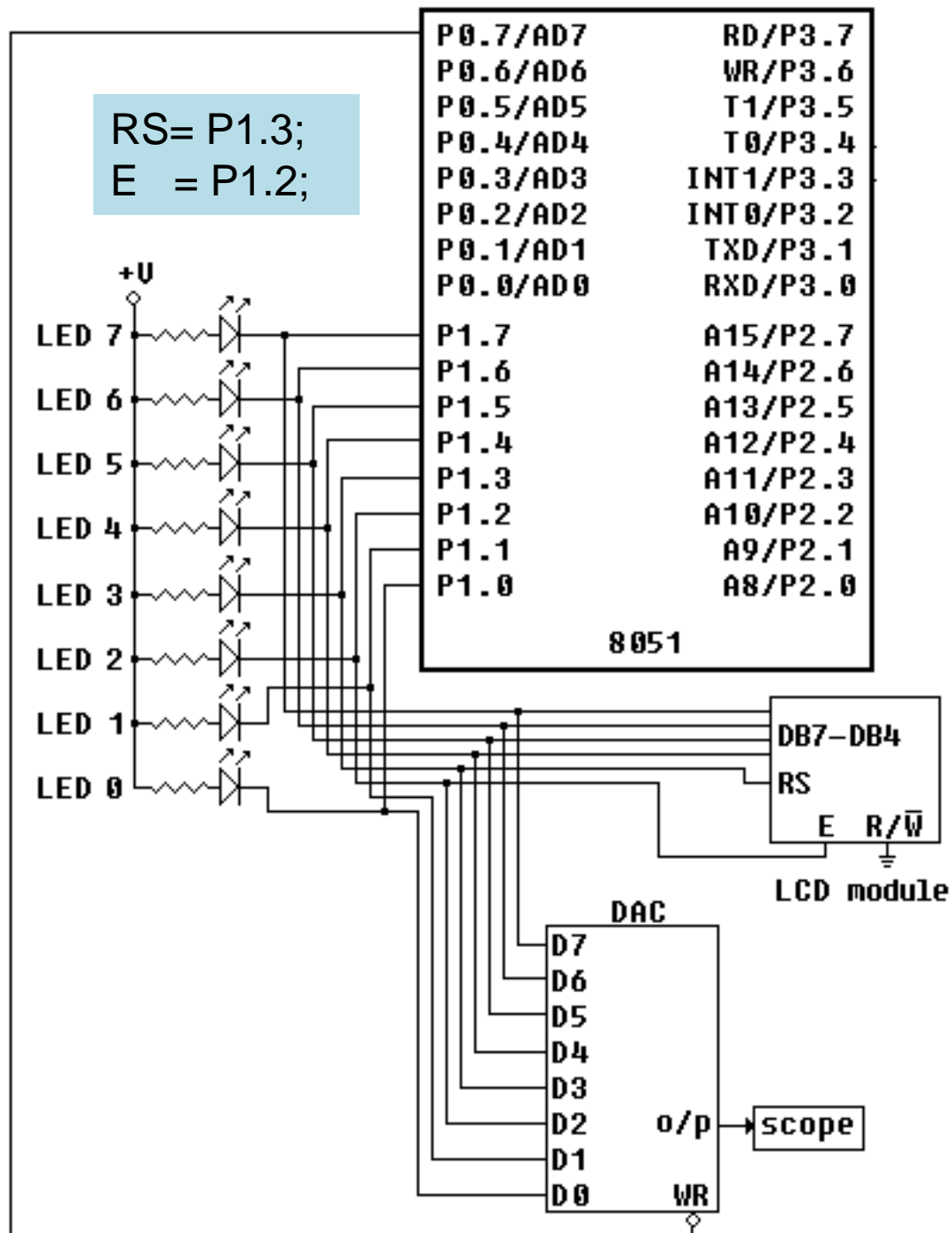
P1  
.6

P1  
.5

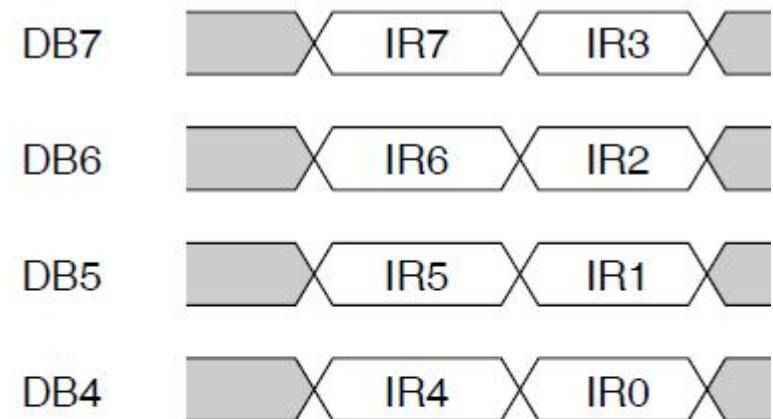
P1  
.4

Execution Time  
(max) (when  $f_{cp}$  or  
 $f_{osc}$  is 270 kHz)

Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	Execution Time (max) (when $f_{cp}$ or $f_{osc}$ is 270 kHz)
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
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 $\mu$ 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 $\mu$ 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 $\mu$ s
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 $\mu$ 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 $\mu$ s
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 $\mu$ s
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 $\mu$ s



### Modelo EDSIM51



Instruction register (IR)  
write

Conexão e procedimento de  
escrita nos registradores internos  
do HD44780 que controla o LCD

--

Interface de 4 pinos

## Sequência de Comandos para Inicialização de LCD Interface de 4 pinos

- 1) 0010 ; 001(DL) DL=0: interface 4 bits – informa comunicação por 4 bits  
; ESPERAR CERCA DE 4,5 ms
- 2) Function Set ; 001(DL)\_NFxx  
; 0010\_10xxb (**28H**) DL=0: interface 4 bits; N=1: 2 linhas; F=0: caractere 5x8  
; (opções não usadas -- DL=1: interface 8 bits; N=0: 1 linha; F=1: caractere 5x10)
- 3) Entry Mode ; 0000\_01(I/D)(SH)  
; 0000\_0110 (**06H**) I/D=1: desloca cursor direita ;SH=0: não desloca mens.  
; (opções não usadas -- I/D=0: desloca cursor esquerda; SH=1: desloca mens.)
- 4) Display on ; 0000\_1DCB  
; 0000\_1111 (**0FH**) D=1: display on; C=1: cursor on; B=1: cursor pisca  
; (opções não usadas -- D=0: display off; C=0: cursor off; B=0: cursor não pisca)

# PROGRAMA PRINCIPAL

ORG 0H

CALL INITDSP

;ESCREVE MENSAGEM

MOV DPTR,#MENS

MOV R2,#08H ; SETB P1.3 = RS - ENVIO DE DADO para LCD

CALL WRT

JMP \$

MENS: DB 13,"EEL7030 - LCD"

DELAY:

MOV R0, #50

DJNZ R0, \$

RET



**INITDSP:**   ; inicializar o display  
              ; 001(DL)\_NFxx (function set)  
              ; 0010\_10xxb DL=0: interface 4 bits; N=1: 2 linhas; F=0: caractere 5x8  
              ; E(P1.2)=0 (1-> 0 = escreve); RS = P1.3 (0=comando;1=dado)

**MOV P1,#20H**   ; FUNCTION SET - high nibble = 0010b -- interface de 4 bits

**SETB P1.2**     ; GERA E  
**CLR P1.2**

**CALL DELAY**     ; AGUARDA LCD FICAR PRONTO

; Nibble alto do Function Set é enviado 2x.

**MOV R2,#0H**     ; CLR P1.3 RS - comando vai ser enviado para LCD  
**MOV DPTR,#comando**

**vai1: CALL WRT**     ; escreve dados para o LCD  
**RET**

**comando: DB 03h,28h,06h,0fh** ; nro. de comandos - function set - entry mode -  
display on/off

**WRT:**

**MOV R1,#1H ; DESLOCAMENTO DO COMANDO/MENSAGEM INICIAL**

**MOV A,#0 ; END. DO NRO DE COMANDOS/DADOS**

**MOVC A,@A+DPTR**

**MOV R6,A ; R6=NRO DE COMANDOS/DADOS**

**LOOP:**

**MOV A,R1 ; END. DO PRIMEIRO DADO/COMANDO EM A**

**MOVC A,@A+DPTR**

**MOV B,A ; BYTE A SER ESCRITO EM B**

**ANL A,#0F0H ; APAGA NIBBLE LS**

**ORL A,R2 ; R2 DEVE CONTER RS (0=COMANDO; 8=DADO) ou seja P1.3**

**MOV P1,A ; ENVIA PARA LCD**

**SETB P1.2 ; GERA E**

**CLR P1.2**

**MOV A,B**

**SWAP A ;TROCA NIBBLES MS-LS**

**ANL A,#0F0H ; APAGA NIBBLE LS**

**ORL A,R2 ; SETB se R2 <> 0 (SETB P1.3=RS (DADO))**

**MOV P1,A ; ENVIA PARA LCD**

**SETB P1.2 ; GERA E**

**CLR P1.2**

**CALL DELAY ; AGUARDA LCD ESTAR PRONTO PARA NOVO COMANDO**

**INC R1 ; R1 APONTA PARA PRÓXIMO COMANDO DADO**

**DJNZ R6,LOOP ; VERIFICA SE ÚLTIMO DADO/COMANDO**

**RET**

**END**

```
//Rotinas para a escrita de mensagem no display do Edsim51 em C  
//Para gerar arquivo com a extensão HEX ir em Settings/Linker  
// modelo de memória no crossware: TINY !!!!!
```

```
#pragma TINY
```

```
#include <reg51.h>  
#include <stdio.h>
```

```
#define CMD 0  
#define DADO 8
```

```
void WRITE (char *,char);    /* função que envia caracteres/cmds para display */  
void INITDP (void);         /* função que inicializa o display */
```

```
sbit EN = P1^2;  /* variável global */
```

```
void main (void)  
{  
char x,cadeia[]="testando$";    /* array contendo mensagem */  
INITDP();                      /* chama rotina de inicializacao dos display */  
WRITE (cadeia,DADO);           /* envia caracteres a serem mostrados no display */  
while(1);                      /* necessario para congelar o 8051 */  
}
```

**/\*Funcao para inicializar o display\*/**

**void INITDP(void)**

**{**

**char init[]={0x28,0x06,0x0f,'\$'}; // vetor com caracteres de inicializacao do display**

**// -- function set - entry mode - display on/off**

**char x;**

**P1=0x20;**

**EN=1;**

**EN=0;**

**for (x=0; x<50; x++);           /\* introduz atraso \*/**

**WRITE(init,CMD);               /\* envia demais bytes de inicializacao \*/**

**}**

```

// Funcao para escrever conjunto de caracteres ou comandos no display ; dado (RS=1); comando (RS=0);
void WRITE(char *dados,char RS)
{
char i=0,x,temp1,temp2;
while (*(dados+i) != '$') {
    temp1 = *(dados+i);
    temp2 = (temp1 & 0xf0)|RS;
    P1= temp2;
    EN=1;
    EN=0;
    temp2 = (temp1 << 4)|RS;
    P1= temp2;
    EN=1;
    EN=0;
    for (x=0; x<50; x++);      /* introduz atraso */
    i++;                      /* desloca byte apontado pelo ponteiro */
}
}

```