

# EEL7030 - Microprocessadores



**LCS**

Laboratório de  
Comunicações  
e Sistemas  
Embarcados

**Prof. Raimes Moraes**  
**EEL - UFSC**

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

**As portas de um microcontrolador são úteis para escrever em leds, ler de chaves, como também interagir com periféricos mais complexos, tais como display de cristal líquido (LCD), módulo GPRS (*General Packet Radio Service*) e outros.**

## Endereços da Área Visível de LCDs de 2 linhas



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

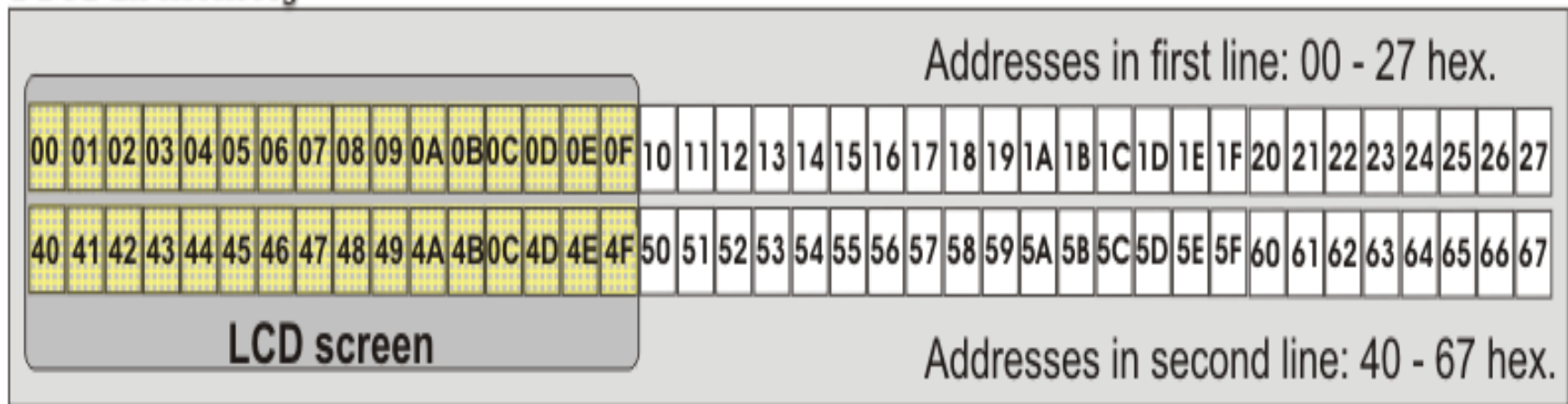
Display	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16						
Line 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	...
Line 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	...

## Endereços da DDRAM de LCDs de 2 linhas

**DDRAM = *Display Data RAM*:** 40 (28H) caracteres em cada linha

**Obs:** Bit 6 do endereço indica a linha (0 ou 1)

DDRAM memory



For  
shift left

01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50

For  
shift right

27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E
67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E

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

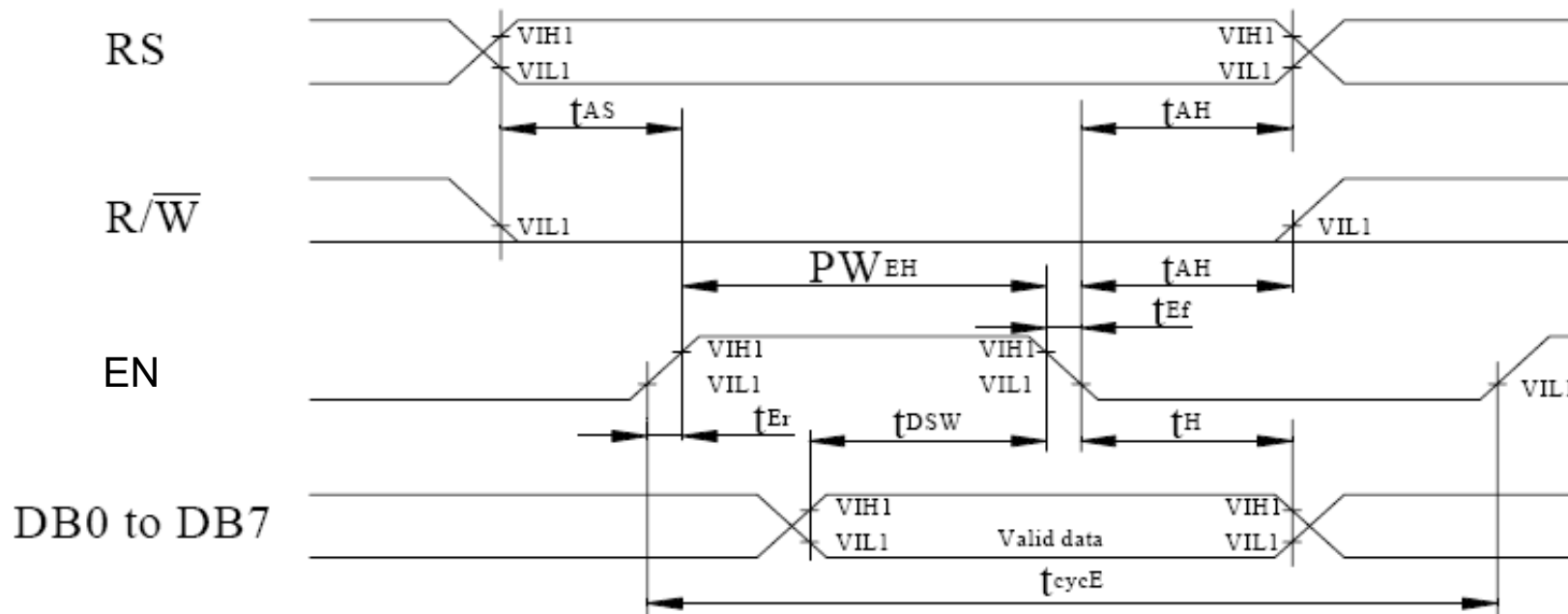
HD44780  
LCD

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

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

--  
Interface de 8 pinos

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



EDSIM51

P1  
.3

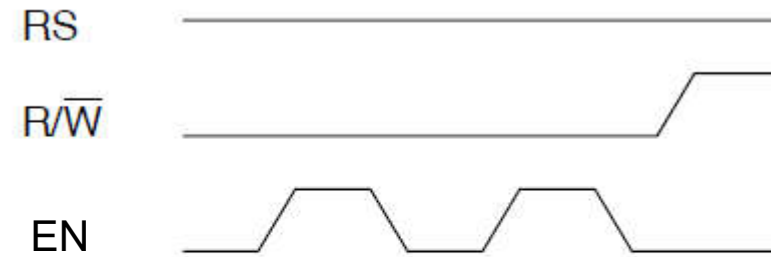
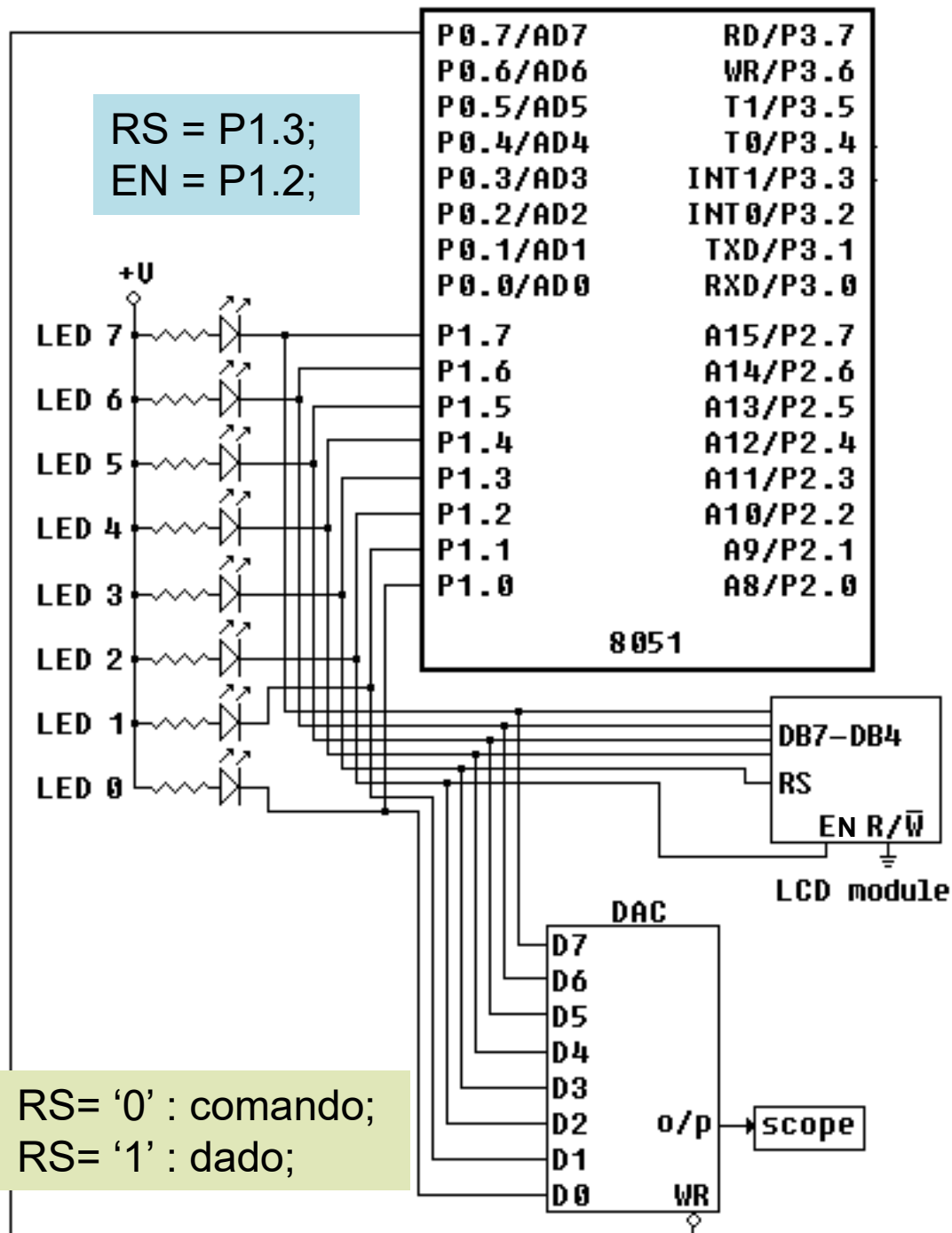
Escreve 4 bits MSB

P1  
.7P1  
.6P1  
.5P1  
.4

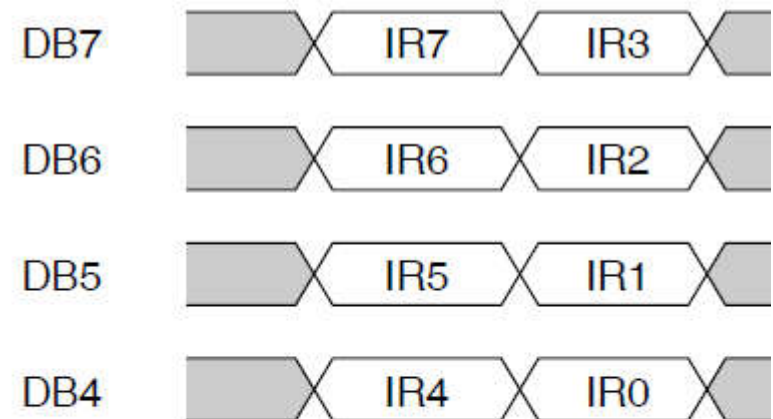
Escreve 4 bits LSB

P1  
.7P1  
.6P1  
.5P1  
.4**DDRAM: Display Data RAM****CGRAM: Character Generator RAM**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

Exemplo de 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 pinos – informa comunicação por 4 trilhas  
; AGUARDAR CERCA DE 4,5 ms
- 2) Function Set ; 001(DL)\_NFxx  
; 0010\_10xxb (**28H**) DL=0: 4 pinos; N=1: 2 linhas; F=0: caractere 5x8  
; (opções não usadas -- DL=1: 8 pinos; N=0: 1 linha; F=1: caractere 5x10)
- 3) 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)
- 4) Entry Mode ; 0000\_01(I/D)S  
; 0000\_0110 (**06H**) I/D=1: desloca cursor direita ; S=0: não desloca mens.  
; (opções não usadas -- I/D=0: desloca cursor esquerda; SH=1: desloca mens.)

**DL:** Data Length  
**N:** Number of display lines  
**F:** character Font

**I/D:** Increment/Decrement DDRAM address  
**S:** Shift display on/off  
(I/D: 0 => right)  
(I/D: 1 => left )

**D:** Display on/off  
**C:** Cursor on/off  
**B:** Blinking cursor on/off



## **Rotina para enviar byte (comando ou caractere) para o HD44780 em barramento de 4 trilhas**

**; Recebe em A, caractere ou comando --- Em R2, 0 se comando; 1 se caractere**

```
WRT2 :  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   EN               ; GERA EN
        CLR    EN

        MOV    A,B              ; RECUPERA BYTE a SER ESCRITO de B
        SWAP   A                ;TROCA NIBBLES MS-LS

        ANL    A,#0F0H          ; APAGA NIBBLE LS
        ORL    A,R2             ; SETB P1.3 (RS) se R2 = RS_1
        MOV    P1,A             ; ENVIA PARA LCD

        SETB   EN               ; GERA EN
        CLR    EN
        CALL   DELAY            ; AGUARDA LCD ESTAR PRONTO PARA NOVO COMANDO
        RET
```

# PROGRAMA PRINCIPAL

```
CS      EQU    P0.7
EN      EQU    P1.2
RS_0    EQU    0
RS_1    EQU    00001000b
Atraso EQU    50H
```

```
ORG 0H
```

```
CLR CS      ; inibe decodificador dos displays de 7 segmentos
CALL INITDSP ; função que inicializa o controlador do display
```

**;ESCREVE MENSAGEM**

```
MOV DPTR,#MENS
MOV R2,# RS_1      ; SETB P1.3 (RS) - ENVIO DE DADO para LCD
CALL WRT
JMP $
```

```
MENS: DB 13,"EEL7030 - LCD"
```

**DELAY:**

```
MOV R0, #Atraso
DJNZ R0, $
RET
```

**INITDSP:** ; inicializa o controlador do display  
 ; 001(DL)\_NFxx = 0010 1000b (function set) = **28H**  
 ; DL=0: interface 4 bits; N=1: 2 linhas; F=0: caractere 5x8  
 ; EN (P1.2): 1-> 0 = escreve; RS (P1.3): 0=comando;1=dado

```
MOV P1,#20H    ; FUNCTION SET - high nibble = 0010b -- interface de 4 pinos
                ;
                ; - RS (P1.3): 0=comando
SETB EN        ; GERA EN
CLR EN
```

**CALL DELAY ; AGUARDA LCD FICAR PRONTO**

**; Nibble alto do Function Set é enviado 2x.**

```
MOV R2,#RS_0 ; CLR P1.3 (RS) - comando será enviado para LCD
MOV DPTR,#comando
```

```

vai1:  CALL    WRT      ; escreve dados no LCD
        RET

```

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

## **Rotina para enviar sequência de bytes (comandos ou caracteres) para o HD44780**

**; Recebe em DPTR, endereço de caracteres ou comandos --- Em R2, 0 se comando; 1 se caractere**

```
WRT:    MOV    A,#0                ; OFFSET de END. do NRO de COMANDOS/DADOS  
        MOVC   A,@A+DPTR  
        MOV    R6,A                ; R6: NRO DE COMANDOS/DADOS  
  
        MOV    R1,#1H              ; DESLOCAMENTO DO COMANDO/MENSAGEM INICIAL  
LOOP:   MOV    A,R1                ; OFFSET de END. de DADO/COMANDO em A  
        MOVC   A,@A+DPTR  
  
        CALL   WRT2  
  
        INC    R1                  ; R1 APONTA PARA PRÓXIMO COMANDO DADO  
        DJNZ   R6,LOOP             ; VERIFICA SE ÚLTIMO DADO/COMANDO  
        RET
```

**Versão do código em C**  
**A ser abordado após estudo de C para 8051**

```

//Rotinas para a escrita de mensagem no display do Edsim51 em C
//Para gerar arquivo com a extensão HEX ir em OPTIONS FOR TARGET1/OUTPUT
// modelo de memória no KEIL: SMALL !!!!! (OPTIONS FOR TARGET1/TARGET)

#pragma SMALL

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

#define CMD 0
#define DADO 8

void WRITE (unsigned char *, unsigned char); // função p/envio dados/cmds para display
void INITDP (void); // função que inicializa o display

sbit CS = P0^7; // variável global
sbit EN = P1^2; // variável global

unsigned char code cadeia[]="testando$"; // array contendo mensagem
unsigned char code init[]={0x28,0x0f,0x06,'$'}; // vetor de inicializacao do display
// -- function set - display on/off - entry mode

void main (void)
{
    CS=0; // inibe displays de 7 segmentos
    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)**

**{**

**unsigned 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(unsigned char *dados, char RS)  
{  
    unsigned char x, temp1;
```

```
while (*dados!= '$') {
```

```
        temp1 = *dados++; // desloca byte apontado pelo ponteiro  
        P1 = (temp1 & 0xf0)|RS;  
        EN=1;  
        EN=0;  
        P1 = (temp1 << 4)|RS;  
        EN=1;  
        EN=0;  
        for (x=0; x<50; x++);           // introduz atraso
```

```
        }           // end of while
```

```
} // end of write
```