

Arquitetura de Computadores

PROF. ISAAC



Display LCD

Códigos de exemplo do 8051

- Subrotina de inicialização do display.

lcd_init

- Subrotina que escreve um caractere no display.

sendCharacter

- Subrotina que posiciona o cursor na linha e coluna.

posicionaCursor

- Subrotina que Limpa o display e posiciona o cursor na primeira linha e primeira coluna.

clearDisplay

Programação

Exemplo 3

Escreva uma rotina que faça o 8051 escrever no Display LCD a palavra FEI centralizada na primeira linha e a palavra Display LCD centralizada na segunda linha.

Observação: Use as sub-rotinas **lcd_init**, **sendCharacter**, **posicionaCursor**, **clearDisplay**.

Exemplo 3

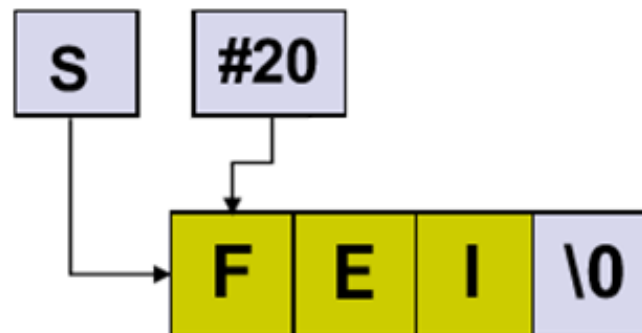
Solução:

Podemos escreve na memória de dados duas String:

FEI e Display LCD

Onde em uma string você tem um endereço inicial e um terminador, nesse caso usamos o **null**.

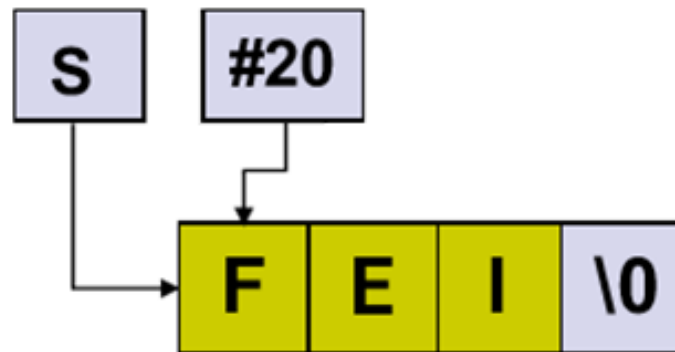
O caractere nulo ou carácter nulo (null character) ou terminador nulo é um caractere da tabela ASCII e do conjunto de caracteres Unicode cujo valor é **0** (zero)



Exemplo 3

Solução:

Abaixo temos a String **FEI** escrita na memória.



; put data in RAM

MOV 20H, #'F'

MOV 21H, #'E'

MOV 22H, #'I'

MOV 23H, #0

;endereço inicial da String FEI

;Marca null no fim da String

Exemplo 3

Solução:

**Podemos escreve na memória de dados as duas Strings:
FEI e Display LCD**

; put data in RAM

```
MOV 20H, #'F' ;endereço inicial da String FEI
MOV 21H, #'E'
MOV 22H, #'I'
MOV 23H, #0    ;Marca null no fim da String
```

; put data in RAM

```
MOV 40H, #'D' ;endereço inicial da String Display LCD
MOV 41H, #'i'
MOV 42H, #'s'
MOV 43H, #'p'
MOV 44H, #'l'
MOV 45H, #'a'
MOV 46H, #'y'
MOV 47H, #' '
MOV 48H, #'L'
MOV 49H, #'C'
MOV 4AH, #'D'
MOV 4BH, #0    ;Marca null no fim da String
```


Exemplo 3

Solução:

Agora criaremos uma subrotina para escrever a String no LCD.

escreveString:

MOV R1, A ; Começa a escrita no endereço de memória apresentado em A
; Inicia a escrita da String no Display LCD

loop:

MOV A, @R1 ; move data pointed to by R1 to A
JZ finish ; if A is 0, then end of data has been reached - jump out of loop
ACALL sendCharacter ; send data in A to LCD module
INC R1 ; point to next piece of data
JMP loop ; repeat

finish:

RET

Exemplo 3

Solução:

Agora no main usaremos as sub-rotinas para escrever as Strings no LCD.

main:

ACALL lcd_init

MOV A, #06h

ACALL posicionaCursor

MOV A, #20h ;endereço inicial de memória da String FEI

ACALL escreveString

MOV A, #42h

ACALL posicionaCursor

MOV A, #40h ;endereço inicial de memória da String Display LCD

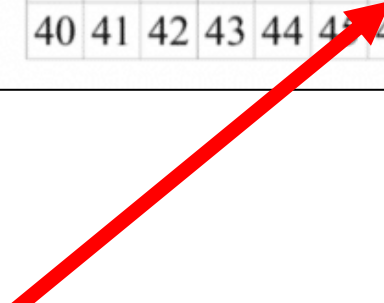
ACALL escreveString

JMP \$

Exemplo 3

Solução:

Character located	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
DDRAM address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F



main:

ACALL lcd_init

MOV A, #06h

ACALL posicionaCursor

MOV A, #20h ;endereço inicial de memória da String FEI

ACALL escreveString

MOV A, #42h

ACALL posicionaCursor

MOV A, #40h ;endereço inicial de memória da String Display LCD

ACALL escreveString

JMP \$

Exemplo 3

Solução:

Character located	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
DDRAM address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

main:

ACALL lcd_init

MOV A, #06h

ACALL posicionaCursor

MOV A, #20h ;endereço inicial de memória da String FEI

ACALL escreveString

MOV A, #42h

ACALL posicionaCursor

MOV A, #40h ;endereço inicial de memória da String Display LCD

ACALL escreveString

JMP \$

Exemplo 3

Solução:

main:

ACALL lcd_init

MOV A, #06h

ACALL posicionaCursor

MOV A, #20h ;endereço inicial de memória da String FEI

ACALL escreveString

MOV A, #42h

ACALL posicionaCursor

MOV A, #40h ;endereço inicial de memória da String Display LCD

ACALL escreveString

JMP \$

Exemplo 4

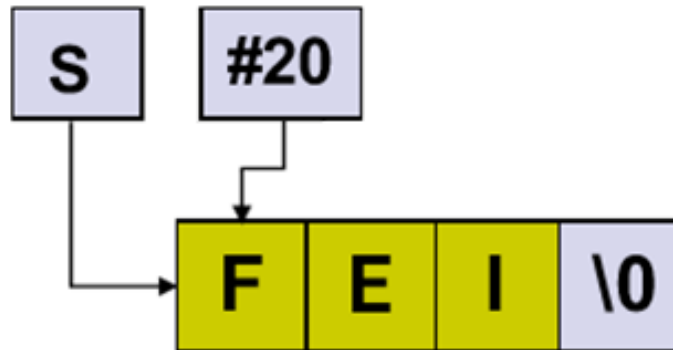
Escreva uma rotina que faça o 8051 escrever no Display LCD a palavra FEI centralizada na primeira linha e a palavra Display LCD centralizada na segunda linha.

Observação: Use as sub-rotinas **lcd_init**, **sendCharacter**, **posicionaCursor**, **clearDisplay**.

Exemplo 4

Solução:

Abaixo temos a String **FEI** escrita na memória do Programa.



; Escrevendo na memória de programa

FEI:

DB "FEI"

DB 0

;caracter null indica fim da String

Display:

DB "Display LCD"

DB 0

;caracter null indica fim da String

Exemplo 4

Solução:

Agora criaremos uma subrotina para escrever a String no LCD.

escreveString:

MOV R2, #0

rot:

MOV A, R2

MOVC A, @A+DPTR ; lê a tabela da memória de programa

ACALL sendCharacter ; send data in A to LCD module

INC R2

JNZ rot ; if A is 0, then end of data has been reached - jump out of loop

RET

Exemplo 4

Solução:

Agora no main usaremos as sub-rotinas para escrever as Strings no LCD.

main:

ACALL lcd_init

MOV A, #06h

ACALL posicionaCursor

MOV DPTR,#FEI

;DPTR = início da palavra FEI

ACALL escreveString

MOV A, #42h

ACALL posicionaCursor

MOV DPTR,#Display

;DPTR = início da palavra Display

ACALL escreveString

JMP \$

Bibliografia

ZELENOVSKY, R.; MENDONÇA, A. Microcontroladores Programação e Projeto com a Família 8051. MZ Editora, RJ, 2005.

Gimenez, Salvador P. Microcontroladores 8051 - Teoria e Prática, Editora Érica, 2010.