

Relatório 1 de SM 2

Programa 1:

Neste programa foram-se utilizadas duas sub-rotinas (IniciaLCD e WriteLCD) com os objetivos de se iniciar o LCD e de se passar os comandos para o LCD, respectivamente e uma interrupção em INT0 (escreve) com o objetivo de se escrever o alfabeto na primeira posição do LCD.

Em relação a sub-rotina passada em classe foram modificadas as posições de EN (de P3.2 para P3.3) e alguns comandos de inicialização, para se evitar conflitos com a interrupção e se permitir a escrita na primeira casa.

RS EQU P3.0

RW EQU P3.1

EN EQU P3.3 ;Mudado de P3.2 para P3.3 para não se ter conflito com interrupção

LCD EQU P1

ORG 0H

JMP Inicio

ORG 3H

LJMP interrup

ORG 100H

Inicio: LCALL IniciaLCD

MOV IE, #81H ;Interrupção na P3.2

MOV TCON, #01H

MOV R1, #65d ;Primeiro caractere do alfabeto

Fim: SJMP Fim

IniciaLCD: PUSH A

MOV A, #00000001b ;Clear Display

ACALL WriteCMD

MOV A, #00000010b ;Cursor na home

ACALL WriteCMD

MOV A, #00001111b ;Display ON/OFF

ACALL WriteCMD

MOV A, #00111100b ;Function set

ACALL WriteCMD

MOV A, #10000001b ;Set DDRAM address

ACALL WriteCMD

MOV A,#80h

ACALL WriteCMD

POP A

RET

WriteCMD: CLR RS ;Modo envia/recebe comando

CLR RW ;Modo escrita

SETB EN ;Prepara cursor para enviar comando ao LCD

MOV LCD,A ;Coloca o comando no barramento externo

CLR EN ;LCD obtém comando no barramento

RET

interrup:LCALL escreve

INC R1 ;Vai para a próxima letra

MOV A,#80h

ACALL WriteCMD

RETI

Escreve:CLR RW ;Modo escrita

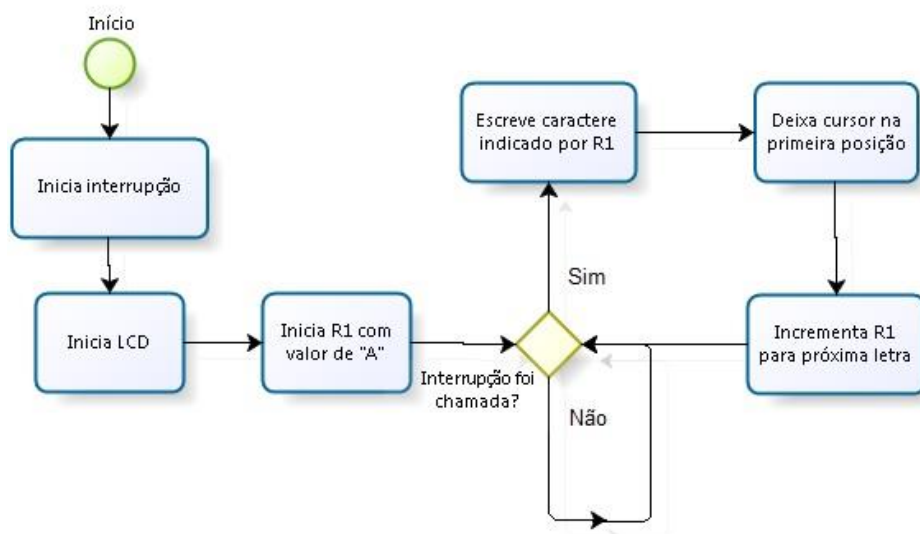
SETB RS ;Modo escrita/novo caractere

SETB EN ;Pulso EN

MOV LCD,R1 ;Coloca caractere no bus

CLR EN ;LCD obtém caractere do bus

Ret



Programa 2:

Neste programa se foram usadas 4 sub-rotinas: IniciaLCD que possui o objetivo de iniciar o LCD, WriteCMD com o objetivo de enviar comandos ao LCD, Atraso para servir de intervalo entre o aparecimento de cada caractere e Escreve que faz a função de escrever o alfabeto no display.

Foram usados registradores para se contar a posição do cursos e se assegurar que não apareceria outros caracteres que não fossem as letras do alfabeto no display, ou seja, parando no Z.



RS EQU P3.0

RW EQU P3.1

EN EQU P3.2

LCD EQU P1

ORG 0

JMP Inicio

ORG 100H

Inicio: LCALL IniciaLCD

MOV R1,#65d ;Primeiro caractere do alfabeto

MOV R4,#16d ;Conta número de colunas antes de ir para a próxima linha

MOV R5,#1d ;Para parar programa no Z

Aqui: LCALL escreve

INC R1 ;Vai para próxima letra

LCALL Atraso

DJNZ R4, Aqui

CJNE R5,#1d, Fim ;Se for segunda passagem vai parar o programa

MOV R4,#11d ;Conta número de colunas antes de parar programa

INC R5

MOV A,#0C0H ;Vai para a próxima linha

ACALL WriteCMD

DEC R1

SJMP Aqui

Fim: SJMP Fim

IniciaLCD: PUSH A

MOV A,#00000001b ;Clear Display

ACALL WriteCMD

MOV A,#00000010b ;Cursor na home

ACALL WriteCMD

MOV A,#00000110b ;Entry mod set

ACALL WriteCMD

MOV A,#00001111b ;Display ON/OFF

ACALL WriteCMD

MOV A,#00011110b ;Shift no Cursor do Display

ACALL WriteCMD

MOV A,#00111100b ;Function set

ACALL WriteCMD

MOV A,#10000001b ;Set DDRAM address

ACALL WriteCMD

POP A

RET

WriteCMD: CLR RS ;Modo envia/recebe comando

CLR RW ;Modo escrita

SETB EN ;Prepara cursor para enviar comando ao LCD

MOV LCD,A ;Coloca o comando no barramento externo

CLR EN ;LCD obtém comando no barramento

RET

ATRASSO: MOV R2, #1h

ATR: MOV R3, #1h

DJNZ R3, \$

DJNZ R2, ATR

RET

Escreve:CLR RW ;Modo escrita

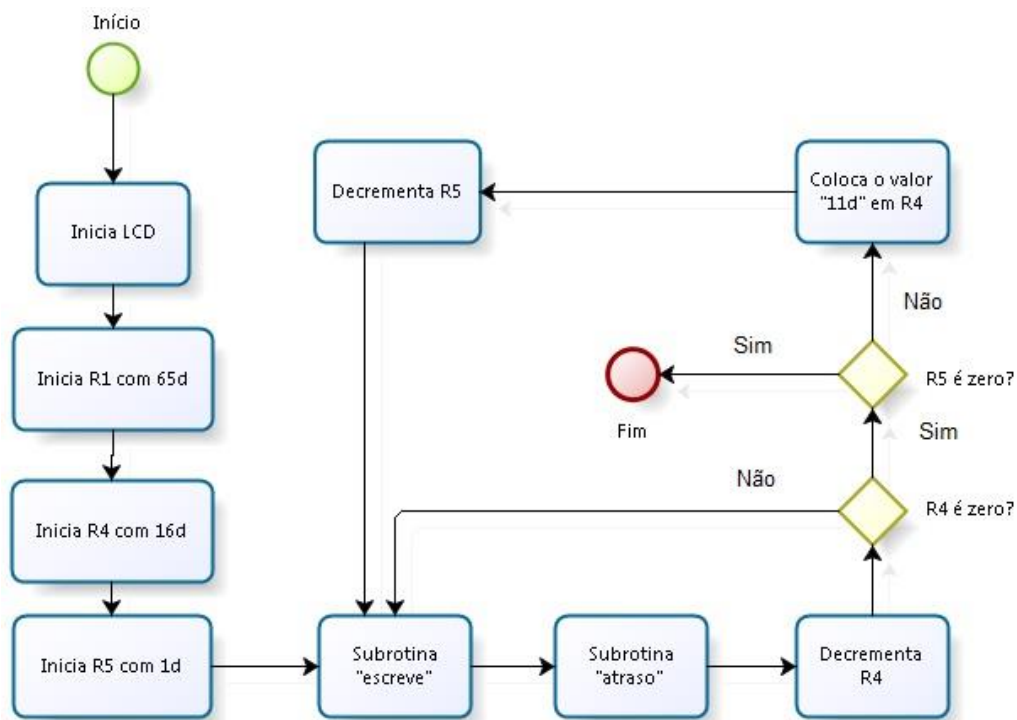
SETB RS ;Modo escrita/novo caractere

SETB EN ;Pulso EN

MOV LCD,R1 ;Coloca caractere no bus

CLR EN ;LCD obtém caractere do bus

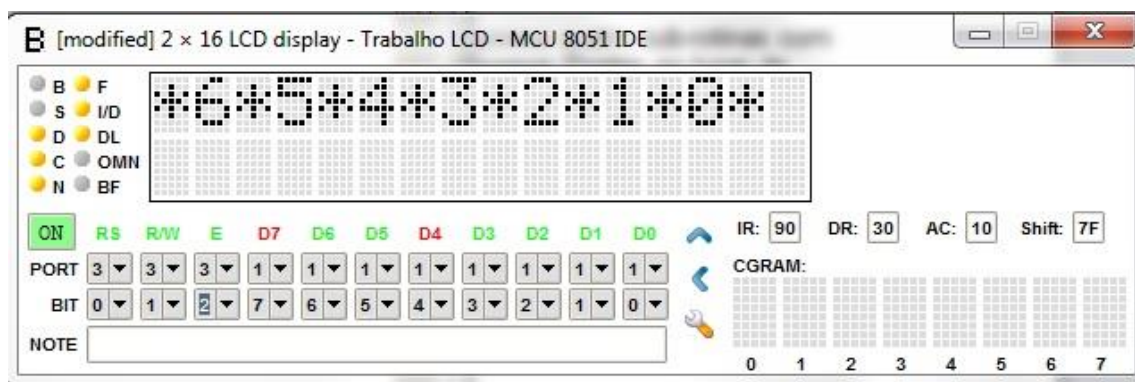
Ret



Programa 3:

Assim como o programa anterior, este também possui quatro sub-rotinas genéricas com as mesmas funções, IniciaLCD, WriteCMD, Atraso e Escreve. Porém, no lugar de escrever o alfabeto nas duas primeiras linhas do display este programa foi feito para fazer uma contagem regressiva de 6 até 0 intercalada de asteriscos, formando o desenho “*6*5*4*3*2*1*0*” na primeira linha do display.

Para alcançar este objetivo se foi utilizado comandos de reposicionamento do cursor, assim como a utilização do datasheet para se encontrar os caracteres necessários.



RS EQU P3.0

RW EQU P3.1

EN EQU P3.2

