

Trabalho Prático Nº 8

Programação de periféricos I/O

Comunicação paralela através dos dispositivos LPTx

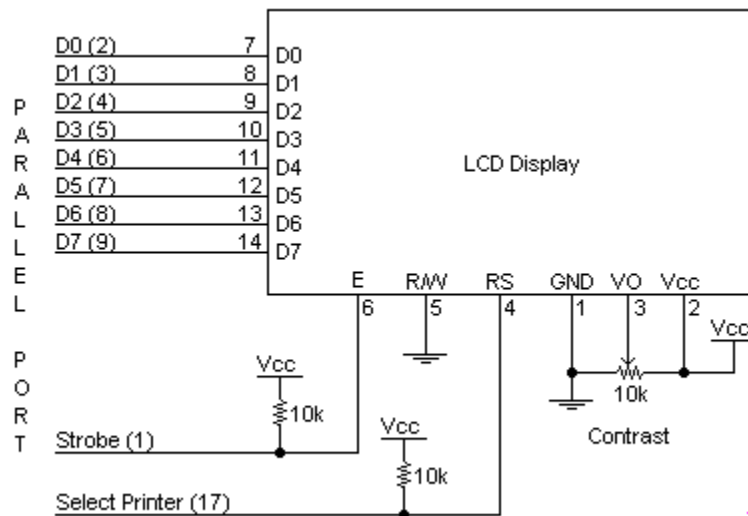
Programação de um display alfanumérico de cristais líquidos

Introdução

Através do interface de comunicação porta paralela (LPT1) de um computador pessoal, pretende-se programar um display alfanumérico de cristais líquidos (LCD).

O LCD utilizado é baseado no controlador HD44780, com uma memória interna organizada numa matriz de 2 linhas por 40 colunas.

A interligação entre o interface LPT1 e o LCD é feito de acordo com o circuito da figura seguinte.



As entradas Enable (E) e Register Select (RS) do LCD estão ligadas, respectivamente, às saídas Strobe (STR) e Select Printer (DSL) do registo de controlo do interface paralelo LPT1. Em alguns PC's estas saídas são em colector aberto, pelo que, por precaução, é necessário incluir uma resistência de pull up de 10K Ω .

A resistência variável de 10 K Ω permite o controlo do contraste do LCD.

A entrada Read/Write(R/W) mantém-se ao nível baixo, já que apenas serão realizadas operações de escrita no LCD.

Programação do LCD

De acordo com o circuito implementado, existem dois tipos de operações para a programação do LCD:

Escrita de instruções (E=1<>0, RS=0, RW=0)

Escrita de dados na memória interna (E=1<>0, RS=1, RW=0)

O LCD procede à leitura do conteúdo do bus de dados (D7-D0) quando se verifica a transição do nível alto para o nível baixo na entrada E (1<>0), devendo manter-se no nível baixo durante um período de tempo. No PC utilizado para o teste deste sistema, este período de tempo é conseguido pela execução do seguinte procedimento:

Procedure Espera;Far;Assembler;

Asm

Push cx

Mov cx,7000

@Ciclo:

Loop @Ciclo

Pop cx

End;

Escrita de instruções:

```

(*Supondo que:
  o endereço base do interface LPT1 é 378h;
  RS=1
  E =1  *)
(* Activar RS=0 *)
mov dx,378h+2 (*Endereço do registo de controlo*)
in al, dx
or al,00001000b (* B3=1 , DSL=0, RS=0 *)
out dx,al
(*Escrita da instrução no registo de dados*)
mov dx,378h (*Endereço do registo de dados*)
mov al,Instrucao
out dx,al
(*Provocar transição E= 1>>0 *)
mov dx,378h+2 (*Endereço do registo de controlo*)
in al, dx
or al,00000001b (* B0=1 , STR=0, E=1>>0 *)
out dx,al
call Espera
(*Provocar transição E= 0>>1 *)
mov dx,378h+2 (*Endereço do registo de controlo*)
in al, dx
and al,11111110b (* B0=0 , STR=1, E=0>>1 *)
out dx,al
(* Desactivar RS=1 *)
mov dx,378h+2 (*Endereço do registo de controlo*)
in al, dx
or al,11110111b (* B3=0 , DSL=1, RS=1 *)
out dx,al

```

Escrita de dados na memória interna:

```

(*Supondo que:
  o endereço base do interface LPT1 é 378h;
  RS=1
  E =1  *)
(*Escrita do dado no registo de dados*)
mov dx,378h (*Endereço do registo de dados*)
mov al,Dado
out dx,al
(*Provocar transição E= 1>>0 *)
mov dx,378h+2 (*Endereço do registo de controlo*)
in al, dx
or al,00000001b (* B0=1 , STR=0, E=1>>0 *)
out dx,al
call Espera
(*Provocar transição E= 0>>1 *)
mov dx,378h+2 (*Endereço do registo de controlo*)
in al, dx
and al,11111110b (* B0=0 , STR=1, E=0>>1 *)
out dx,al

```

As diferentes instruções de programação do LCD estão descritas nos documentos em anexo.

Trabalho a realizar

Pretende-se desenvolver um conjunto de subrotinas que permitam a programação do LCD. Deve consultar a informação técnica sobre o interface paralelo LPTx e o dispositivo LCD que se encontra em anexo.

Subprogramas a implementar:

```

Procedure GetAddressLPT(N:Byte);Far;Assembler;
(* Em função do número N (1,2,3,4) do interface paralelo, determina o
endereço base do interface, colocando-o na variável global
ADR_LPT(Word) *)

Procedure STRHigh;Far;Assembler;
(* Coloca o sinal STR(LPTx)=1 = E (LCD)=1
B0=0 no Registo de Controlo de LPTx *)

Procedure STRLow;Far;Assembler;
(* Coloca o sinal STR(LPTx)=0 = E (LCD)=0
B0=1 no Registo de Controlo de LPTx *)

Procedure DSLHigh;Far;Assembler;
(* Coloca o sinal DSL(LPTx)=1 = RS (LCD)=1
B3=0 no Registo de Controlo de LPTx *)

Procedure DSLLow;Far;Assembler;
(* Coloca o sinal DSL(LPTx)=0 = RS (LCD)=0
B3=1 no Registo de Controlo de LPTx *)

Procedure LCDWrite(D:Byte);Far;Assembler;
(* Escreve o valor D no LCD
Assume E=1
D => Registo de Dados de LPTx
E = 1 >> 0
Espera 3 ms
E = 0 >> 1 *)

Procedure LCDInit;Far;Assembler;
(* Inicializa o LCD com as seguintes instruções
Function Set - 0011xxxx(Interface=8bits)
Function Set - 0011xxxx(Interface=8bits)
Function Set - 0011xxxx(Interface=8bits)
Function Set - 001111xx(Interface=8bits;2Linhas;Matriz5x10)
Display ON/OFF - 00001000(Display OFF)
Display ON/OFF - 00001110(Display ON;Cursor ON;Blink OFF)
Entry Mode Set - 00000110(I/D = Incrementação; Cursor Movement)
*)

Procedure LCDShiftDisplayLeft;Far;Assembler;
(* Desloca o conteúdo da memória interna do LCD de uma posição para a
esquerda *)

Procedure LCDShiftDisplayRight;Far;Assembler;
(* Desloca o conteúdo da memória interna do LCD de uma posição para a
direita *)

Procedure LCDClear;Far;Assembler;
(* Apaga o conteúdo do LCD *)

Procedure LCDGotoLC(L,C:Byte);Far;Assembler;
(* Desloca cursor para a linha L (0,1) e coluna C(0..39) *)

Procedure LCDWriteStr(S:String);Far;Assembler;
(* Escreve a string S no LCD a partir da actual posição do cursor *)

```

Programa de teste:

```
(* Assumindo variáveis globais
```

```
    ADR_LPT:Word;  
    S0,S1:String;  
*)  
Begin  
    GetAddressLPT(1);  
    DSLHigh;  
    STRHigh;  
    LCDInit;  
    Readln;  
    Write(' Linha 0?'); Readln(S0);  
    Write(' Linha 1?'); Readln(S1);  
    LCDGotoLC(0,0);  
    LCDWriteStr(S0);  
    Readln;  
    LCDGotoLC(1,0);  
    LCDWriteStr(S1);  
    Readln;  
    Repeat  
        LCDShiftDisplayLeft;  
        Delay(500);  
    Until Keypressed;  
    Readln;  
    Repeat  
        LCDShiftDisplayRight;  
        Delay(500);  
    Until Keypressed;  
    Readln;  
    LCDClear;  
    Readln;  
End.
```