

## Trabalho Prático Nº 9

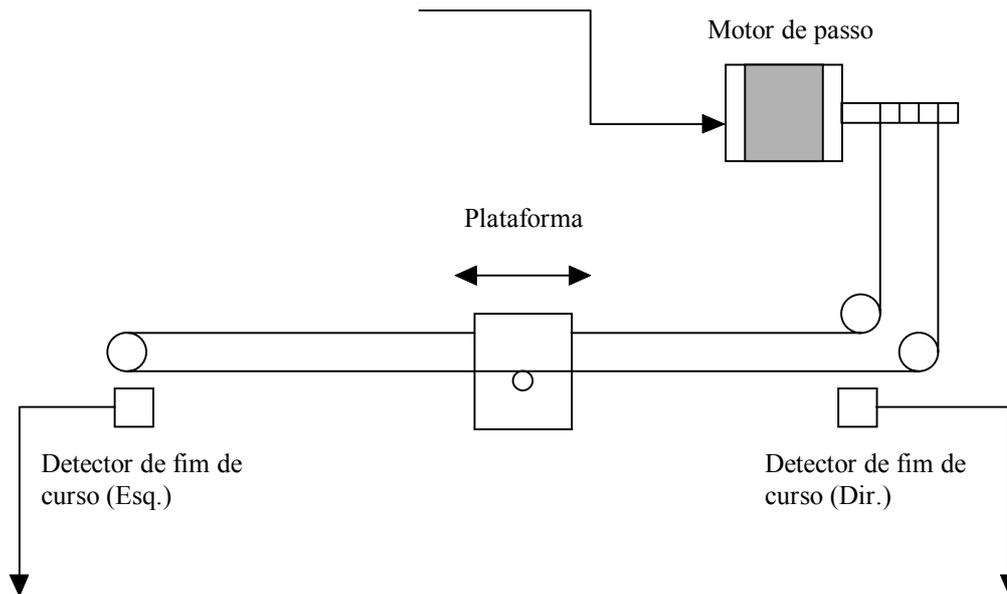
Programação de periféricos I/O

Comunicação paralela através dos dispositivos LPTx

Controlo de um motor de passo

### Introdução

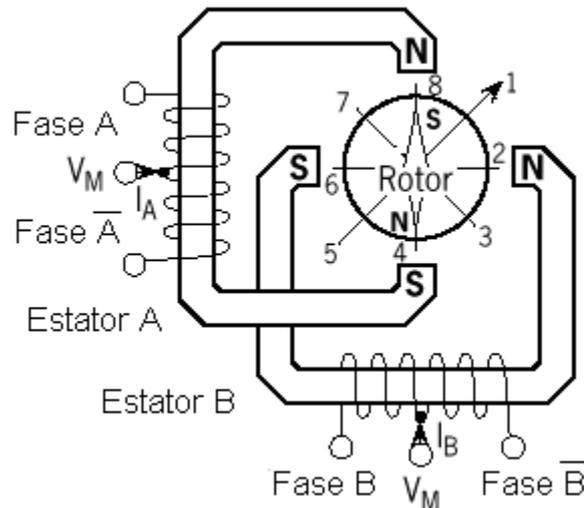
Através do interface de comunicação porta paralela (LPT1) de um computador pessoal, pretende-se controlar o sistema mecânico apresentado na figura.



Através de um sistema de roldanas, o movimento rotacional do motor de passo é transformado em movimento de translação aplicado a uma pequena plataforma. Os detectores de fim de curso são dispositivos digitais que indicam se a plataforma móvel se encontra numa das extremidades.

### Princípio de funcionamento do motor de passo

O motor de passo utilizado é um motor unipolar constituído por 4 bobinas (4 fases) que envolvem um material electromagnético (estator). O rotor (parte móvel do motor) é constituído por um material magnetizado de forma permanente. Quando uma ou mais bobinas são submetidas a uma tensão de alimentação, o campo magnético induzido no estator provoca um movimento de rotação no rotor até se atingir um ponto de equilíbrio.



### ***Controlo do movimento rotacional***

O controlo do movimento rotacional do rotor é conseguido pela aplicação sequencial da tensão de alimentação nas bobinas. Neste tipo de motor, é possível aplicar os seguintes modos de funcionamento (“Stepping modes”):

#### “Wave Drive”

Neste modo de funcionamento, apenas umas das bobinas se encontra alimentada. Ao aplicarmos de forma sequencial, uma tensão de alimentação nas fases  $A \rightarrow B \rightarrow \bar{A} \rightarrow \bar{B}$ , o rotor roda, passo a passo, das posições  $8 \rightarrow 2 \rightarrow 4 \rightarrow 6$ .

#### “Full Step Drive”

Neste modo de funcionamento, duas bobinas são simultaneamente alimentadas. Ao aplicarmos, simultaneamente, de forma sequencial, uma tensão de alimentação nas fases  $AB \rightarrow \bar{A}B \rightarrow \bar{A}\bar{B} \rightarrow A\bar{B}$ , o rotor roda, passo a passo, das posições  $1 \rightarrow 3 \rightarrow 5 \rightarrow 7$ .

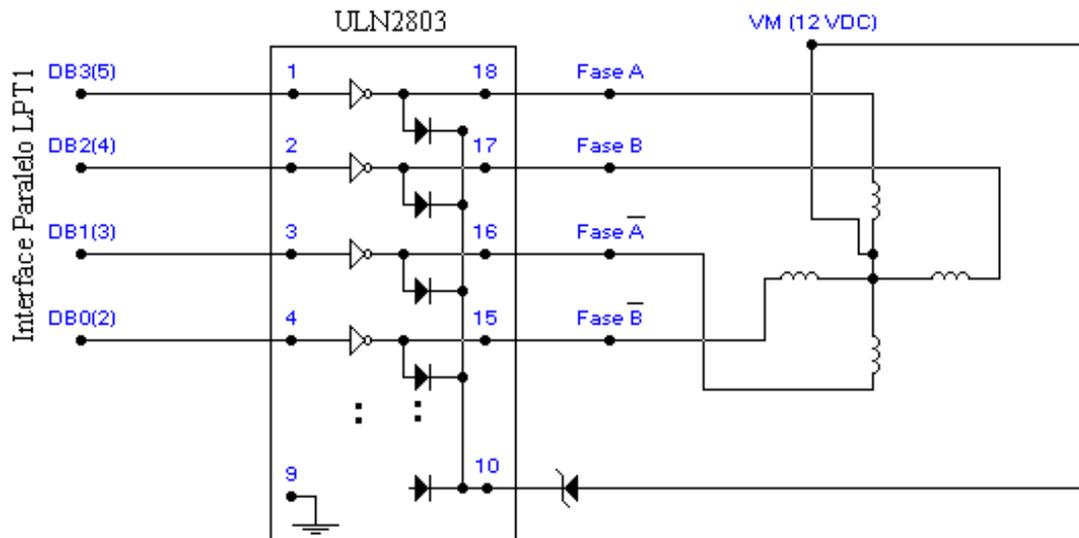
#### “Half Step Drive”

Este modo de funcionamento combina os dois modos anteriores permitindo uma rotação de meio passo. Ao aplicarmos, simultaneamente, de forma sequencial, uma tensão de alimentação nas fases  $AB \rightarrow B \rightarrow B\bar{A} \rightarrow \bar{A} \rightarrow \bar{A}\bar{B} \rightarrow \bar{B} \rightarrow A\bar{B} \rightarrow A$ , o rotor roda, passo a passo, das posições  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$ .

### **Alimentação do motor de passo**

A figura seguinte descreve a interligação entre as saídas DB3-DB0 do interface paralelo LPT1 às bobinas do motor de passo. É necessária a utilização do circuito integrado ULN2803 para a condução (“drive”) das intensidades de corrente absorvidas pelas bobinas. O ULN2803 permite a condução de intensidades de corrente até 500mA, valor suficiente para as características das bobinas do motor de passo do sistema. As entradas 1-4 são compatíveis TTL e permitem controlar o estado ligado/desligado das bobinas, de acordo com a tabela seguinte:

Entrada 1-4	Bobine (Fase)
1	Ligada
0	Desligada



### Modos de funcionamento

De acordo com o esquema anterior, o controlo do modo de funcionamento do motor de passo poderá ser obtido por aplicação sequencial e temporizada de valores lógicos nas linhas DB3-DB0 do interface LPT1, conforme se descreve na tabela seguinte:

Modo	DB3-DB0	Fases (Bobines ligadas)
Wave Drive	1000 (8h)	A
	0100 (4h)	B
	0010 (2h)	$\bar{A}$
	0001 (1h)	$\bar{B}$
Full Step Drive	1100 (Ch)	AB
	0110 (6h)	$B\bar{A}$
	0011 (3h)	$\bar{A}\bar{B}$
	1001 (9h)	$A\bar{B}$
Half Step Drive	1100 (Ch)	AB
	0100 (4h)	B
	0110 (6h)	$\bar{A}B$
	0010 (2h)	$\bar{A}$
	0011 (3h)	$\bar{A}\bar{B}$
	0001 (1h)	$\bar{B}$
	1001 (9h)	$A\bar{B}$
	1000 (8h)	A

### Controlo de velocidade

A velocidade de rotação é facilmente controlada pela introdução de um tempo de espera entre duas sequências consecutivas. Para este trabalho deverá ser utilizado o procedimento a seguir apresentado, devendo ser utilizado o valor mínimo de  $T=1$  para obter a velocidade máxima.

```

Procedure Espera (T:Word) ;Near;Assembler;
asm
  push ax
  push cx
  xor ax,ax
@While:
  cmp ax,T

```

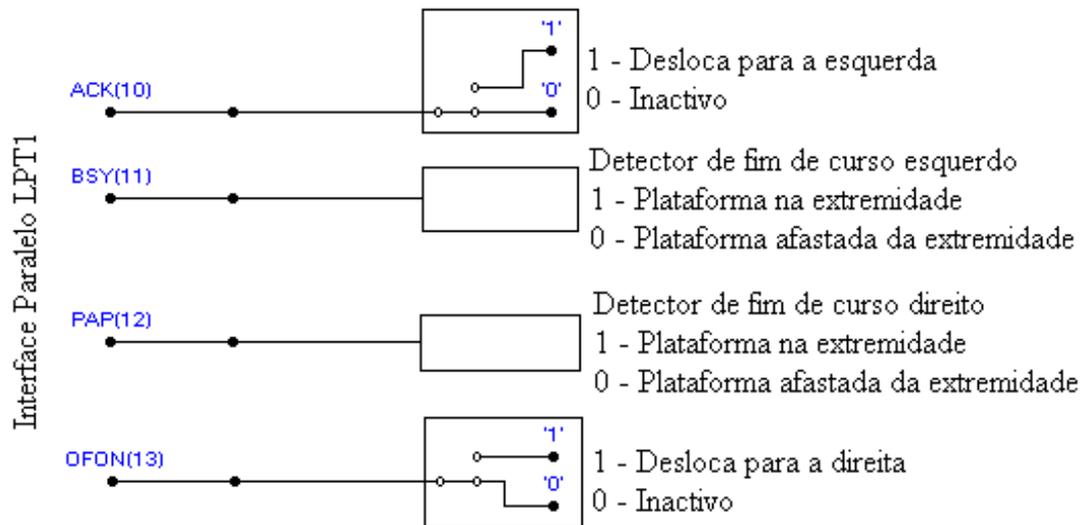
```

je @EndWhile
mov cx,15000
@Ciclo:
loop @Ciclo
inc ax
@EndWhile:
pop cx
pop ax
end;

```

### Controlo do sistema

Para além das linhas DB3-DB0 do interface LPT1, o sistema utiliza um conjunto de entradas adicionais do interface LPT1, de acordo com o esquema da figura seguinte:



As entradas ACK e OFON permite o deslocamento da plataforma móvel para a esquerda ou direita respectivamente. As entradas BSY e PAP permitem detectar se a plataforma móvel se encontra na extremidade esquerda ou direita respectivamente.

### Trabalho a realizar

Pretende-se desenvolver um conjunto de subrotinas que permitam o controlo do sistema.

#### **Subprogramas a implementar:**

Considerando as seguintes declarações:

```

Uses Crt;
Const
WaveStep:Array[1..8] of Byte = ($08,$04,$02,$01,$08,$04,$02,$01);
FullStep:Array[1..8] of Byte = ($0C,$06,$03,$09,$0C,$06,$03,$09);
HalfStep:Array[1..8] of Byte = ($0C,$04,$06,$02,$03,$01,$09,$08);
Var SteppingMode: Pointer;
    StepIndex:ShortInt;
    ADR_LPT:Word;
    TEspera:Word; (*Controlo da velocidade*)
    NoSteps:Word;

```

```

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 SetSteppingMode(M:Byte); Far; Assembler;
(* Coloca o ponteiro SteppingMode a apontar para o início de umas das
tabelas WaveStep, FullStep, HalfStep, em função do modo de
funcionamento M.
M=0 > WaveStep
M=1 > FullStep
M=2 > HalfStep
Coloca a variável global StepIndex a zero
*)

```

```

Function StepMoveRight:Byte;Far;Assembler;
(* Devolve o próximo código binário a enviar para as linhas DB3-DB0
Actualiza a variável StepIndex em conformidade (Incrementação)
*)

```

```

Function StepMoveLeft:Byte;Far;Assembler;
(* Devolve o próximo código binário a enviar para as linhas DB3-DB0
Actualiza a variável StepIndex em conformidade (Decrementação)
*)

```

```

Procedure HomeRight;Far;Assembler;
(* Desloca a plataforma móvel para a extremidade direita *)

```

```

Procedure HomeLeft;Far;Assembler;
(* Desloca a plataforma móvel para a extremidade esquerda
Coloca a variável NoSteps a zero *)

```

```

Function StepControl:Byte;
(*
Verifica o estado das entradas ACK e OFON
Se ACK activa então desloca um passo para a esquerda
Se OFON activa então desloca um passo para a direita
Verifica se a plataforma já se encontra numa das extremidades
Incrementa/Decrementa a variável NoSteps
Invoca subprograma Espera com T=TEspera
Devolve o conteúdo do registo de estado da interface LPTx
*)

```

### ***Programa principal:***

```

Begin
  ClrSCr;
  TEspera:=1;
  GetAddressLPT(1);
  SetSteppingMode(0);
  HomeRight;
  HomeLeft;
  SetSteppingMode(1);
  Repeat
  GotoXY(1,1);
  Writeln(StepControl);
  Writeln(NoSteps);
  Until Keypressed;
  Readln;
  SetSteppingMode(2);
  Repeat
  GotoXY(1,1);
  Writeln(StepControl);
  Writeln(NoSteps);
  Until Keypressed;
  Readln;
End.

```