

Instituto Politécnico de Viseu
Departamento de Engenharia Electrotécnica

Sistemas de Tempo Real

Monta Cargas



Ricardo Antunes	N.º	3737
Ricardo Sobral	N.º	2814
Paulo Gouveia	N.º	3242

Índice

1. Introdução	2
2. Estrutura	3
3. Hardware	8
• Placa para controlo de um só motor:	8
Funcionamento:	9
• Placa para controlo de dois motores:	10
• Placa para controlo de electroíman:	11
Funcionamento:	11
• Fonte de alimentação:	12
• Ventilador:	12
• Caixa:	13
• Interligações:	14
4. Apresentação do trabalho final	16
5. Software	17
• Programa Elaborado.....	19
6. Considerações finais	24
7. Bibliografia:	26

1. Introdução

No âmbito da cadeira Sistema de Tempo Real, pretendeu-se desenvolver um trabalho onde o conceito de sistemas em tempo real fosse demonstrado na prática.

Um sistema em tempo real não é nada mais do que um sistema “normal” com a particularidade de responder em tempo útil, ou seja, responder num determinado tempo (tempo esse que varia de sistema para sistema) a uma acção de controlo efectuada por um controlador (ex.: CPU).

Este tipo de trabalho tem como fundamental objectivo tentar aproximar os alunos á realidade ,como futuros engenheiros e desenvolver individualmente capacidades de criar novas soluções quando as já existentes estão ultrapassadas.

O trabalho desenvolvido consiste na construção de um monta cargas (toda a estrutura, mecânica e eléctrica) e no desenvolvimento do respectivo hardware e software para controlar todos os seus movimentos.

Para tal, tivemos que ter em consideração o material necessário para a sua construção e o modo de obter esse mesmo material, sem que isso implicasse gastos elevados para o departamento.

Por este motivo foi considerado o aspecto económico e tentou-se realizar a construção do monta cargas recorrendo a material que já não tinha utilidade como por exemplo impressoras, fotocopiadoras, restos de madeira, ...

2. Estrutura

Material utilizado.

- Madeira (Contraplacado);
- Parafusos;
- Cola;
- Fio condutor;
- Junções;
- Motores de passo;
- Electroímã;
- Varias peças retiradas de impressoras e fotocopiadora.

Como se chegou à estrutura final do monta cargas.



A estrutura do monta cargas foi feita de madeira (contraplacado). Nesta fase tivemos a ajuda de um colega do curso de Engenharia de Madeiras, o qual nos cedeu a madeira (madeira essa que já não tinha qualquer utilidade) e a cortou de acordo com as medidas estabelecidas.

De seguida procedeu-se a montagem das peças de madeira, para a qual se utilizou cola e parafusos de forma a dar a esta estrutura uma maior resistência.

Inicialmente pensamos apenas serem necessários 3 motores de passo:

- Um para deslocar o electroímã para cima/baixo;



- Um para deslocar o electroímã para a direita/esquerda;



- Um para deslocar toda a estrutura par a frente/traz;



No entanto verificou-se que no deslocamento para a frente/trás das rodas, a parte onde não havia motor ficava imóvel, devido ao facto de um só motor não ser suficiente para fazer deslocar ambas as partes. Sendo então necessário acrescentar um outro motor de passo como mostra a seguir.



Foi com um pouco de imaginação que assentamos a estrutura em 8 rodas as quais permitem o movimento para a frente/traz.

A ligação das rodas ao motor de passo foi feita através de uma corrente.



Nesta figura vemos o sistema que permite o deslocamento direita/esquerda. Retiramos a chamada cabeça da impressora para realizarmos esta operação.



Por fim vemos o electroímã, o qual permite agarrar/desagarrar os objectos.



3. Hardware

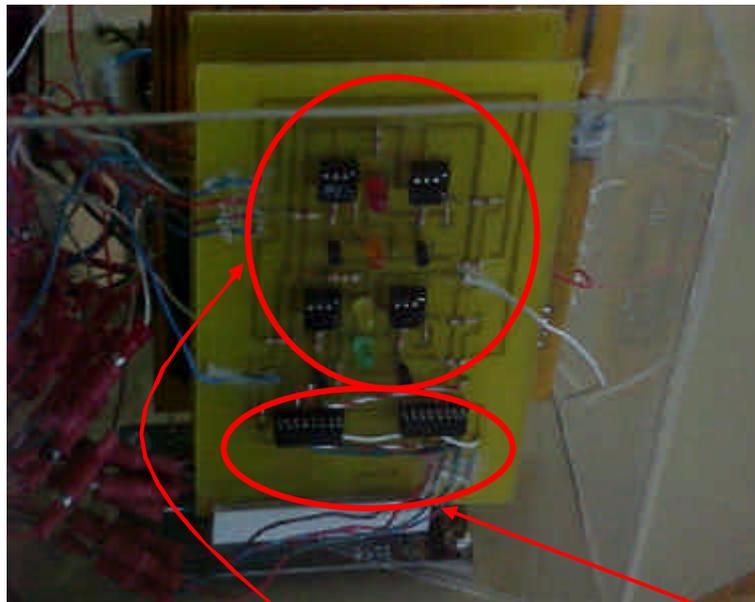
Dado que a corrente de saída na porta paralela é muito baixa relativamente ao valor necessário para os motores de passo foi necessário a criação de hardware para elevar o valor desta corrente.

Com este fim elaborámos três placas para elevar a corrente de 4 motores de passo, duas delas idênticas, as quais controlam apenas um motor, e uma outra, que vai controlar dois motores de passo. Elaborámos ainda uma outra placa para controlar o electroímã.

Uma vez que a porta paralela não tem qualquer tipo de protecção internamente, estas placas têm também a função de dar uma certa protecção externa a esta porta.

Para alimentar estas placas utilizamos uma fonte de alimentação de um PC, a qual é alimentada por sua vez externamente a 230V. Utilizámos ainda um ventilador de forma a refrigerar a caixa onde se encontra a fonte de alimentação, bem com as placas.

- **Placa para controlo de um só motor:**

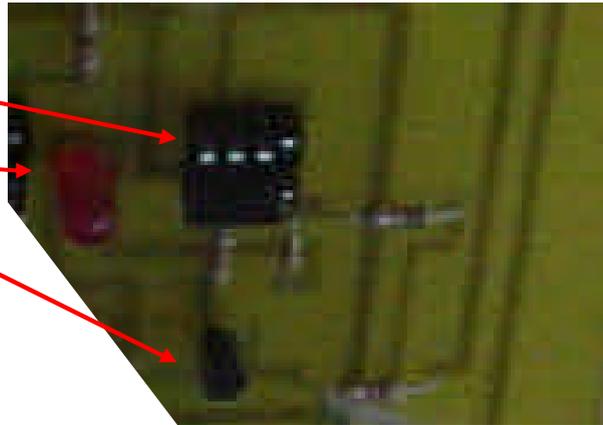


Esta placa é constituída essencialmente por quatro “grupos” de componente, sendo os componentes de cada grupo iguais para os quatro grupos, por um diodo de zener, e por dois integrados (ULN2003) colocados em paralelo um com o outro.

Cada um destes “grupos” de componentes tem como função controlar um dos quatro bits necessários ao funcionamento do motor de passo.

Assim sendo cada grupo é constituído por:

- Um acoplador óptico (4N32);
- Um led.
- Um transístor (BC 547);
- Quatro resistências;



Funcionamento:

O sinal (bit) é recebido da porta paralela, atravessa uma resistência, (a qual vai provocar uma queda de tensão), e é aplicado à base do transístor, sai pelo colector deste já amplificado, atravessa uma outra resistência, provocando outra queda de tensão e é então aplicado ao acoplador óptico. O sinal sai então do acoplador óptico e vai ser aplicado simultaneamente aos integrados que têm por função amplificar o sinal e estabelecer a ligação com os motores de passo, e ao led. Este led tem como função, única e simplesmente, de dar a indicação que o sinal está a ser dado ao “grupo” no qual ele está inserido, permitindo-nos assim saber qual o sinal (bit) que está a ser aplicado em cada instante ao motor de passo, para que em caso de ocorrência de algum erro possamos actuar de forma correctiva.

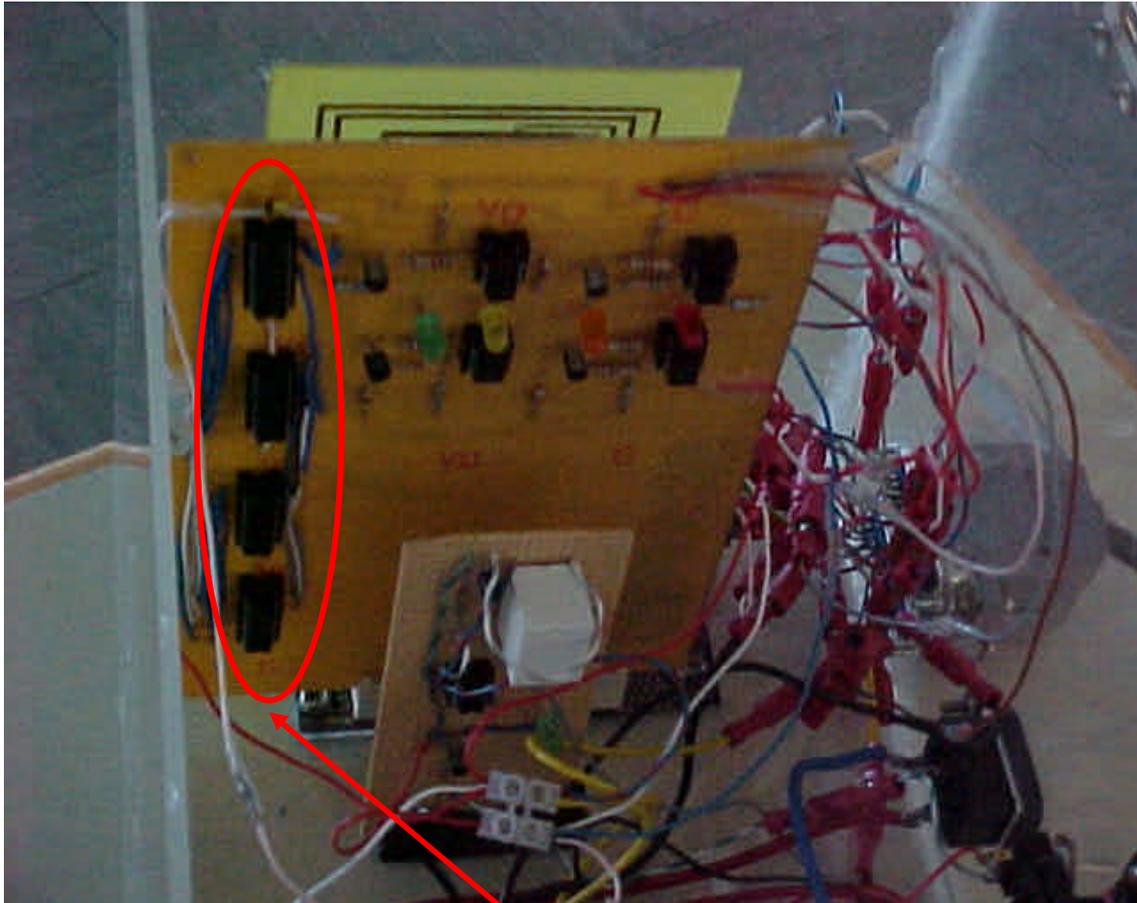
O díodo de zener presente em cada placa limita o sinal que vai alimentar os integrados de amplificação do sinal.

O sinal depois de amplificado é então aplicado ao MPP.

Dado que cada motor funciona através da aplicação de quatro sinais, estes são aplicados à entrada de cada um dos “grupos” alternadamente, mas de forma sequencial.

É de salientar ainda que os integrados são alimentados a partir da fonte de alimentação a 12Ve que os acopladores ópticos funcionam como elementos de protecção.

- **Placa para controlo de dois motores:**



A única diferença desta placa relativamente à anterior reside no facto desta ter intercalados em paralelo com os outros dois integrados mais dois deforma a compensar a corrente necessária para os dois motores que esta placa vai controlar.

O princípio de funcionamento desta placa é o mesmo da anterior.

- **Placa para controlo de electroíman:**

Esta placa vai controlar a alimentação do electroíman e é constituída por:

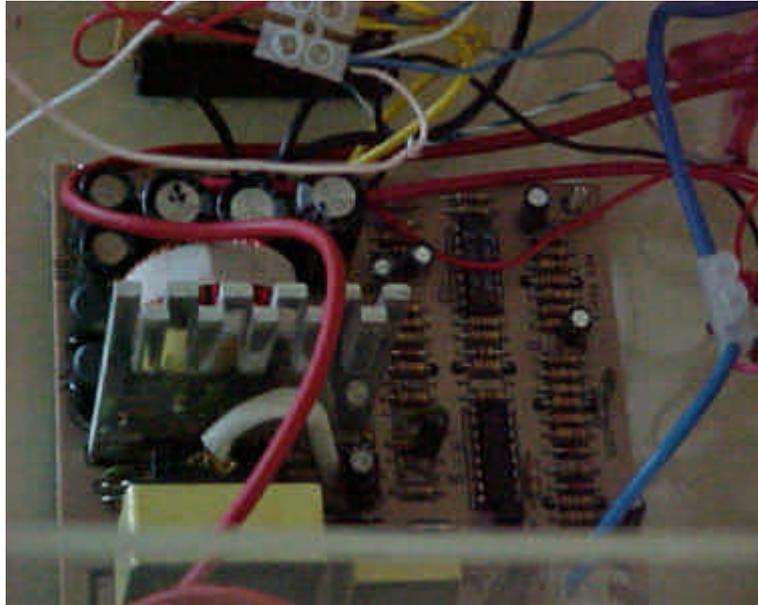
- Um acoplador óptico (4N32);
- Um led.
- Dois transístor (BC 547);
- Quatro resistências;
- Um diodo de zener (12 V)



Funcionamento:

O principio de funcionamento desta placa é similar ao das outras três, a diferença fundamental reside no facto de nesta não utilizarmos um integrado para amplificar o sinal, mas sim um relê, o qual está alimentado a 12V, e quando lhe é aplicado o sinal atraca de forma a alimentar o electroíman.

- **Fonte de alimentação:**



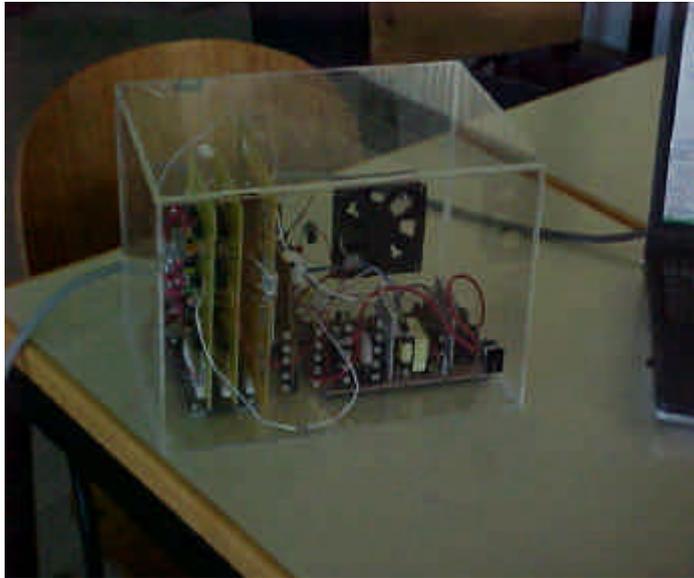
Esta fonte é alimentada a 230 V e permite-nos alimentar as os integrados das placas a 12 V.

- **Ventilador:**



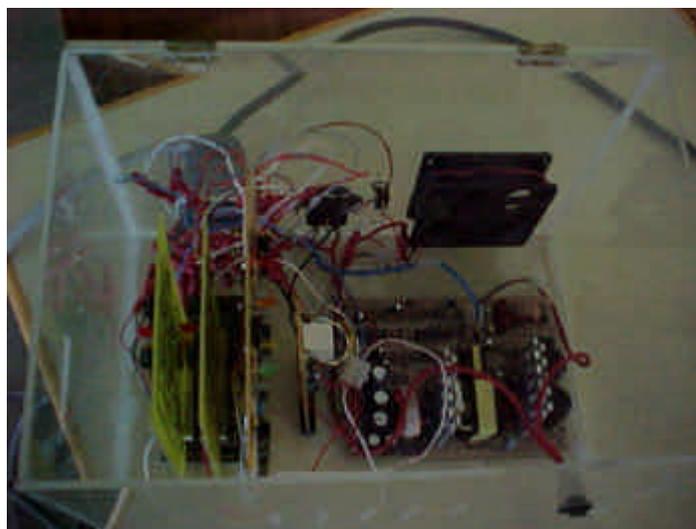
Este ventilador permite a refrigeração do ar dentro da caixa

- **Caixa:**



Esta caixa é feita em acrílico, perfurada de forma a facilitar a circulação do ar, na superfície de trás encontram-se fixas as portas de entrada e de saída de dados, a ficha para a alimentação da fonte, e ainda o ventilador, estando este fixo interiormente. Na superfície da frente encontra-se um interruptor on/off par podermos ligar/desligar a alimentação. Interiormente encontram-se alojadas as quatro placas, três para o controlo dos motores e uma para o electroímã, a fonte de alimentação e o ventilador.

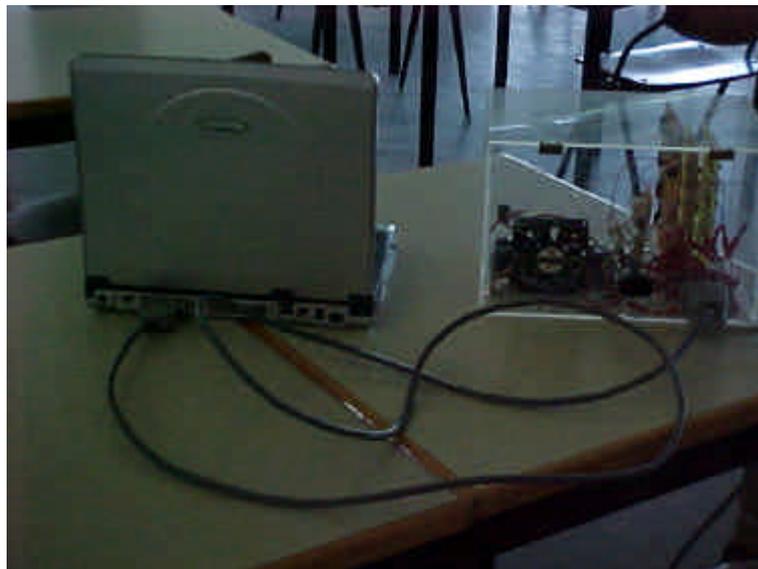
Esta caixa permite-nos ainda um fácil acesso ao seu interior, bastando par tal abria a tampa, que se encontra na parte superior desta fixa por duas dobradiças.



- **Interligações:**

A interligação dos diversos sistemas é realizada da seguinte forma:

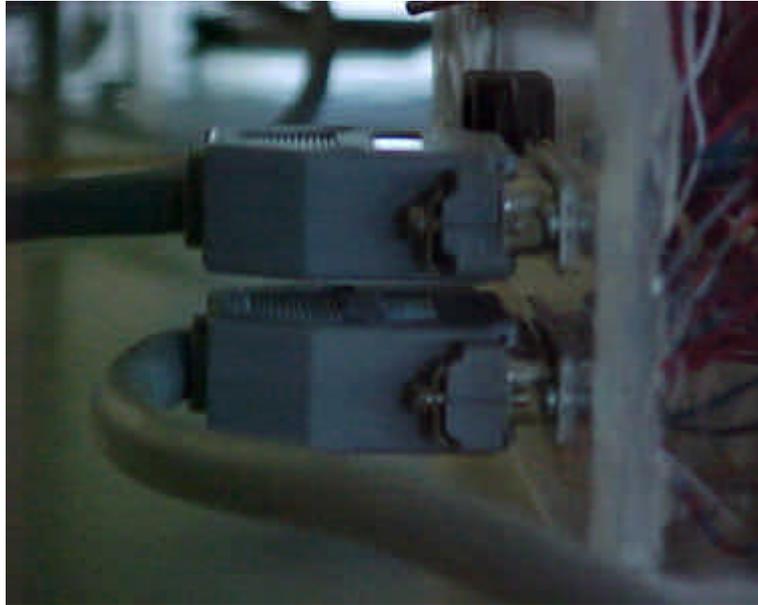
A ligação da porta paralela do PC com a caixa efectua-se através de fichas de 25 pinos, sendo estas ligadas por um cabo:



A ligação da entrada da caixa com as placas, bem como das placas com a saída da caixa, é conseguida através de ligadores, os quais nos permitem em caso de avaria retirara as placas para fora da caixa de forma a facilitar o reparo da avaria:



A saída da caixa é feita, tal como a entrada por uma ficha de 25 pinos:



A interligação desta ficha de saída com o monta cargas é feita através de um cabo, ligando num barramento, do qual por sua vez se liga aos respectivos motores e ao electroíman.



4. Apresentação do trabalho final.



Podemos ver na figura as três partes que essencialmente constituem o projecto:

- O software, ou seja o programa em Ada 95
- A parte de hardware que se encontra dentro da caixa de acrílico
- A estrutura do monta cargas

5. Software

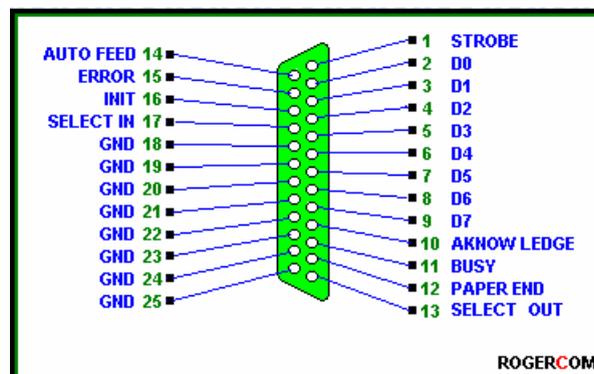
Para efectuar o controlo dos movimentos do Monta Cargas, utilizamos como linguagem de programação o ADA95.

O uso desta linguagem deve-se ao facto de ser um tipo linguagem desenhada especificamente para sistemas de tempo real embutidos, ou seja, para sistemas que realizam acções de controlo. Este sistema que realiza a acção de controlo não é mais do que um componente de um sistema a controlar.

Sendo um dos objectivos do trabalho controlar o Monta Cargas utilizando a porta paralela, tivemos que realizar um estudo prévio acerca da porta, uma vez que, seria necessário conhecer os portos que efectuassem comunicação com o exterior e seus respectivos endereços.

Após uma breve pesquisa, verificamos que temos disponíveis 12 pinos de saída na porta paralela sendo 8 referentes ao porto de dados e 4 ao porto de controlo.

Nome	Endereços LPT1	Descrição
Registro de Dados	378h	Envia um byte para a impressora
Registro de Status	379h	Ler o Status da impressora
Registro de Controle	37Ah	Envia dados de controle para a impressora



Os pinos 2,3,4,5,6,7,8,9 são referentes ao porto de dados e os 1,14,16,17 ao porto de controlo.

Relativamente ao porto de controlo foi necessário ter em atenção que alguns dos pinos eram activos a zero. Tal facto criou alguma confusão na elaboração do programa, quando era pretendido

enviar para o exterior uma determinada sequência. Os pinos activos a zero são o 1,14,17 e a um apenas o 16.

Sendo o objectivo controlar quatro motores de passo (embora dois deles estejam em paralelo, leva a que o seu controlo seja feito como se tratasse de apenas um) e sendo necessário dar uma sequência de impulsos de 4 bits para cada um deles, atribuímos ,sem qualquer razão específica, os seguintes bits da porta paralela a cada motor/movimento realizado:

D0,D1,D2,D3 – Movimento Sobe/Desce Íman

D4,D5,D6,D7, - Movimento Carro Superior (Direita/esquerda)

1,14,16,17 – Movimento Frente/Trás

Solucionados estes problemas, iniciamos a elaboração do programa.

Uma das características deste tipo de linguagem é a concorrência. Por este motivo utilizamos pacotes já pré-definidos de modo a facilitar a comunicação com a porta paralela. Foram utilizados os pacotes *IO_PORTS* e *INTERFACES*.

- Programa Elaborado

```
with Io_Ports, Interfaces, Ada.Integer_Text_Io, Ada.Text_Io;
use Io_Ports, Interfaces, Ada.Integer_Text_Io, Ada.Text_Io;

procedure Escrever is

    Tecla: Integer;
    Register: Unsigned_16 := Unsigned_16(16#378#);
    Data: Unsigned_8;

    -----//-----
    ---Movimento do imam (Sobe / Desce)-----
    --Utilizando os portos de dados (D0;D1;D2;D3)--
    -----da porta Paralela-----

    procedure Desce_Imam is

    begin
        Put_Line("Inicio de descida");
        New_Line;
        for I in 1..4 loop
            Put_Line("Descida");
            Data := Unsigned_8(2#00000001#);
            Write_Io_Port(Register, Data);
            Enable_Interrupts;
            Data := Unsigned_8(2#00000010#);
            delay 0.1;
            Disable_Interrupts;
            Write_Io_Port(Register, Data);
            Enable_Interrupts;
            Data := Unsigned_8(2#00000100#);
            delay 0.1;
            Disable_Interrupts;
            Write_Io_Port(Register, Data);
            Enable_Interrupts;
            Data := Unsigned_8(2#00001000#);
            delay 0.1;
            Disable_Interrupts;
            Write_Io_Port(Register, Data);
            Enable_Interrupts;
            delay 0.1;
        end loop;
        Put_Line("Fim de descida");
        New_Line;
    end Desce_Imam;

    procedure Sobe_Imam is

    begin
        Put_Line("Inicio de Subida");
        New_Line;
        for I in 1..4 loop
            Put_Line("Subida");
```

```
Data:=Unsigned_8(2#0001000#);
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
Data:=Unsigned_8(2#00000100#);
delay 0.1;
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
Data:=Unsigned_8(2#00000010#);
delay 0.1;
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
Data:=Unsigned_8(2#00000001#);
delay 0.1;
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
delay 0.1;
end loop;
Put_Line ("Fim de subida");
New_Line;
end Sobe_Imam;

-----//---
--Movimento do Carro Superior (Direita / Esquerda)---
--Utilizando os portos de dados (D4;D5;D6;D7)--
-----da porta Paralela-----

procedure Movimento_Direita is

begin
Put_Line("Inicio Movimento Direita");
New_Line;
for I in 1..12 loop
Put_Line("Direita");
Data:=Unsigned_8(2#00010001#);
Write_Io_Port(Register,Data);
Enable_Interrupts;
Data:=Unsigned_8(2#00100001#);
delay 0.01;
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
Data:=Unsigned_8(2#01000001#);
delay 0.01;
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
Data:=Unsigned_8(2#10000001#);
delay 0.01;
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
delay 0.01;
end loop;
```

```
Put_Line ("Fim Movimento Direita");  
New_Line;  
end Movimento_Direita;
```

```
procedure Movimento_Esquerda is
```

```
begin
```

```
Put_Line ("Inicio Movimento Esquerda");  
New_Line;  
for I in 1..12 loop  
    Put_Line("Esquerda");  
    Data:=Unsigned_8(2#1000001#);  
    Disable_Interrupts;  
    Write_Io_Port(Register,Data);  
    Enable_Interrupts;  
    Data:=Unsigned_8(2#0100001#);  
    delay 0.01;  
    Disable_Interrupts;  
    Write_Io_Port(Register,Data);  
    Enable_Interrupts;  
    Data:=Unsigned_8(2#0010001#);  
    delay 0.01;  
    Disable_Interrupts;  
    Write_Io_Port(Register,Data);  
    Enable_Interrupts;  
    Data:=Unsigned_8(2#0001001#);  
    delay 0.01;  
    Disable_Interrupts;  
    Write_Io_Port(Register,Data);  
    Enable_Interrupts;  
    delay 0.01;  
end loop;  
Put_Line ("Fim Movimento Esquerda");  
New_Line;  
end Movimento_Esquerda;
```

```
-----//-----  
--Movimento da monta cargas para a Frente e para Tras--  
--Utilizando os portos de controlo da porta paralela--  
----- com o endereço 37AH-----
```

```
procedure Movimento_Frente is
```

```
Register:Unsigned_16:=Unsigned_16(16#37A#);
```

```
begin
```

```
Put_Line("Inicio Movimento Frente");  
New_Line;  
for I in 1..4  
    loop  
        Put_Line("Frente");  
        Data:=Unsigned_8(2#1010#);  
        Write_Io_Port(Register,Data);
```

```
Enable_Interrupts;
Data:=Unsigned_8(2#1001#);
delay 0.1;
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
Data:=Unsigned_8(2#1111#);
delay 0.1;
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
Data:=Unsigned_8(2#0011#);
delay 0.1;
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
delay 0.1;
end loop;
Put_Line ("Fim Movimento Frente");
New_Line;
end Movimento_Frente;

procedure Movimento_Tras is

Register:Unsigned_16:=Unsigned_16(16#37A#);

begin
Put_Line ("Inicio Movimento Tras.");
New_Line;

for I in 1..4 loop
Put_Line("Tras");
Data:=Unsigned_8(2#0011#);
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
Data:=Unsigned_8(2#1111#);
delay 0.1;
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
Data:=Unsigned_8(2#1001#);
delay 0.1;
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
Data:=Unsigned_8(2#1010#);
delay 0.1;
Disable_Interrupts;
Write_Io_Port(Register,Data);
Enable_Interrupts;
delay 0.1;
end loop;
Put_Line ("Fim Movimento Tras");
New_Line;
end Movimento_Tras;
```

```
----- Programa principal -----  
-----////-----  
  
begin  
  
Put_Line ("Inicio do programa");  
Put_Line ( " ");  
  
begin  
  loop  
    Put_Line("          ESCOLHA OPCAO          ");  
    New_Line;  
    Put_Line("          Tecla-1 = Descer Imam          ");  
    Put_Line("          Tecla-2 = Sobe Imam          ");  
    Put_Line("          Tecla-3 = Movimento_Frente          ");  
    Put_Line("          Tecla-4 = Movimento_Tras          ");  
    Put_Line("          Tecla-5 = Movimento_Esquerda          ");  
    Put_Line("          Tecla-6 = Movimento_Direita          ");  
    New_Line;  
    Put_Line("          Tecla-0 = Sair do Programa          ");  
    New_Line;  
    Put_Line("          Escolha Tecla....          ");  
    New_Line;  
    Get(Tecla);  
  
    if Tecla = 1 then  
      Desce_Imam;  
    end if;  
  
    if Tecla= 2 then  
      Sobe_Imam;  
    end if;  
  
    if Tecla = 3 then  
      Movimento_Frente;  
    end if;  
  
    if Tecla= 4 then  
      Movimento_Tras;  
    end if;  
  
    if Tecla= 5 then  
      Movimento_Direita;  
    end if;  
  
    if Tecla=6 then  
      Movimento_Esquerda;  
    end if;  
  
    if Tecla=0 then  
      Put_Line ("fim de programa");  
      exit;  
    end if;  
  
  end loop;  
end;  
end Escrever;
```

6. Considerações finais

Na elaboração deste trabalho deparamo-nos com inúmeras dificuldades.

Logo de início enfrentamos o problema relativo à aquisição do material, pois tivemos que recorrer a diferentes fontes para o adquirir.

Ultrapassada esta fase inicial, montamos toda a estrutura do sistema e iniciamos a concepção do hardware.

Nesta segunda fase surgiram-nos novos problemas uma vez que a componente electrónica não era o forte do grupo.

Os problemas encontrados nesta fase referem-se à concepção das placas (sendo as condições para esta prática muito precárias), fase de teste das placas (componentes já avariados, ligações defeituosas...). Além destas dificuldades foi necessário efectuar um estudo prévio acerca do funcionamento dos componentes, pois alguns eram desconhecidos.

Posto isto iniciamos a programação para controlar o sistema.

Nesta área, além da dificuldade encontrada por ser uma nova linguagem de programação, tivemos que encontrar na placa paralela os pinos que realizavam a comunicação com o exterior, a respectiva sequência para os dados serem enviados correctamente e os endereços de cada porto utilizado.

Com o decorrer do trabalho e com a tentativa de solucionar todos estes problemas obtivemos novos conhecimentos tanto a nível de hardware como de software.

Contudo, não podemos deixar de agradecer ao Técnico afecto ao Laboratório de Física Eng.º Olas dada a sua prestação na aquisição do material necessário e ao docente da cadeira Eng.º Moisés pelo acompanhamento na elaboração do trabalho

A estrutura do projecto encontra-se presentemente guardada num armário no Laboratório de Instalações Eléctricas.



7. Bibliografia:

- Apontamentos do docente da cadeira se Sistemas de Tempo Real
- Informações retiradas de paginas Web
- Manuais de programação em Ada95



Trabalho realizado por:

Ricardo Antunes	N.º 3737
Ricardo Sobral	N.º 2814
Paulo Gouveia	N.º 3242