

# **AUTOMAÇÃO DO PROCESSO DE TECELAGEM DE ELETRODOS MULTIFILARES**

Maria Isabel da Costa Bandeira, Germano Pinto Guedes e Ademarlaudo França Barbosa

## Índice

### I – Detecção de Raios-X

1.1 - Detetores de Raios-X a Gás _____	1
1.2 - Eletrodos Multifilares _____	1
1.3 - Processo de Tecelagem de Eletrodos Multifilares _____	3

### II - Automação do Processo de Tecelagem de Eletrodos Multifilares

2.1 - Sistema Mecânico _____	4
2.2 - Monitoramento e Alinhamento do Fio _____	4
2.3 - Proteção de Fim de Curso _____	5
2.4 - Movimento do Cubo _____	6
2.5 - Movimento do Suporte do Fio _____	7
2.5.1 - Especificações _____	7

### III – Circuito Eletrônico

3.1 - Controle de Motor de Passo Bipolar _____	9
3.1.1 – Especificações de Funcionamento _____	10
3.2 - Controle de Corrente para Motor DC _____	11
3.3 - Circuito Gerador de Pulsos _____	11
3.4 - Polarização dos Sensores _____	12

### IV - Interface Paralela

4.1 - A Porta Paralela do PC _____	13
4.2 - Configuração da Porta Paralela na Automação do Sistema _____	15

### V – Programação

5.1 - Rotina “Iniciar Tecelagem” _____	17
5.2 - Tabela de Estados _____	18
5.3 - Código Fonte do Programa _____	19

**ANEXO A** – Esquema Elétrico

**ANEXO B** - Diagrama Esquemático

## I – Detecção de Raios-X

### 1.1 - Detetores de Raios-X a Gás

A capacidade de ionizar gases foi um aspecto dos raios-x identificado desde os tempos de sua descoberta. Apesar de terem sido desenvolvidos vários tipos de detetores para diferentes aplicações dos raios-x, os detetores a gás foram historicamente os mais utilizados, e continuam sendo desenvolvidos por responderem satisfatoriamente a diversos requisitos considerados conflitantes em muitas aplicações.

Num detetor a gás a idéia básica é coletar em eletrodos a carga elétrica liberada por ionização após a passagem dos fótons por um meio gasoso (figura1). Se  $E_f$  é a energia do fóton e  $E_i$  a energia média requerida por ionização, um número médio  $E_f/E_i$  de pares elétron-íon são liberados por fóton absorvido. Estes pares tendem a se neutralizar e se recombinar com as moléculas do próprio gás. Para a detecção efetiva do fóton faz-se necessário o estabelecimento de uma diferença de potencial elétrico,  $V_0$ , de modo que os pares sejam separados e coletados em eletrodos.

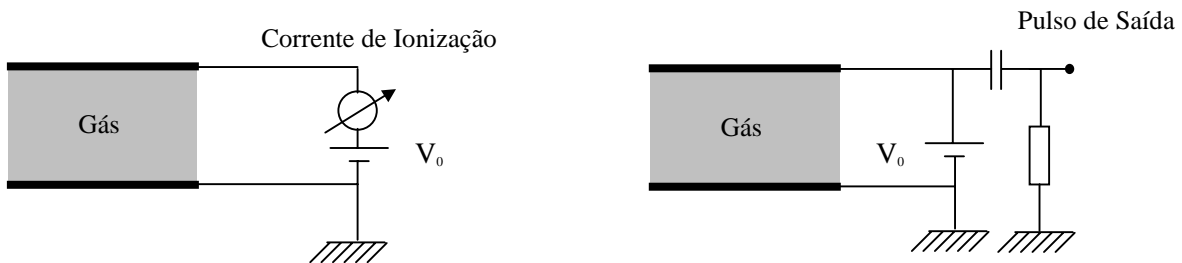


Figura 1: Circuito esquemático de detetores a gás em modo de câmara de ionização (à esquerda) e de contador proporcional (à direita).

### 1.2 - Eletrodos Multifilares

No final dos anos 60 foi proposto um novo tipo de detetor a gás, no qual a posição de incidência do fóton era estimada a partir de uma medida de tempo. Utilizando um fio resistivo como eletrodo captador de elétrons com capacitores em suas extremidades, tem-se para cada fóton detectado uma constante de tempo RC para carregar os capacitores que depende da posição sobre o fio onde foi absorvido o fóton. Essa idéia foi estendida com a introdução de linhas de retardo, onde a posição de

incidência do fóton é diretamente associada ao tempo de propagação do pulso elétrico através da linha. ( figura2 ).

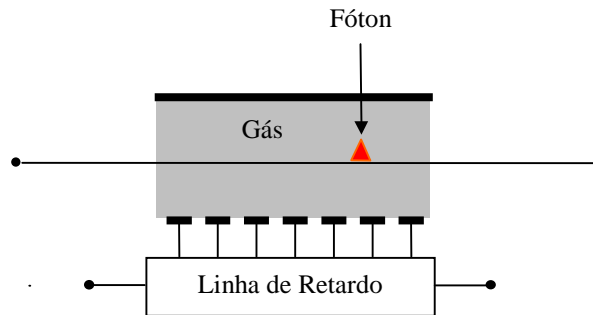


Figura 2: Esquema de decodificação da posição de um fóton a partir do tempo de transmissão de sinais por linha de retardo.

Este sistema é ainda mais aprimorado quando se agrupa um conjunto de fios distribuídos em um plano. Assim, com duas linhas de retardo se pode decodificar as coordenadas de posição de uma partícula ionizante em duas dimensões [Ref.: A. F. Barbosa. *Nuclear Instruments and Methods in Physics Research*. A371, 368-374 (1996)]. Para desenvolver detetores bi-dimensionais utiliza-se portanto um arranjo contendo basicamente um plano de fios paralelos dispostos entre dois planos condutores, de forma a constituir um dispositivo que permita o processo de detecção e localização de partículas (figura 3).

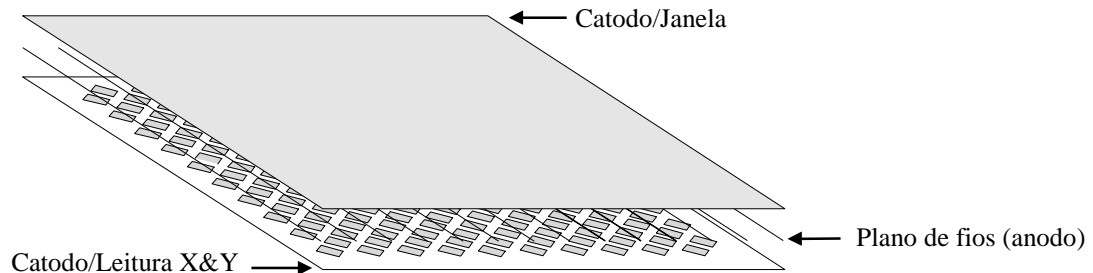


Figura 3: Esquema de um contador multifilar

Este plano de fios, denominado Eletrodo Multifilar, é formado por uma matriz de fios de tungstênio-rênio banhados a ouro com diâmetro que pode variar de 5 a 100  $\mu\text{m}$  dependendo do tipo de detector. Eles são soldados paralelamente em uma placa de circuito impresso com espaçamento típico de 1 ou 2 mm.

### 1.3 - Processo de Tecelagem de Eletrodos Multifilares

Antes de chegar à placa de circuito impresso final dos eletrodos multifilares os fios passam por um processo de tecelagem que define com precisão o espaçamento entre eles e a sua tensão mecânica. Diferentes tipos de sistemas mecânicos podem ser utilizados para executar este processo. Utilizamos aqui um sistema formado por uma estrutura cúbica giratória que tem seu eixo posicionado sobre duas hastes verticais fixas em uma base sólida (figura 4). Nesta base apoia-se também um fuso por onde desliza um suporte, no qual se fixa a bobina do fio. O processo consiste em esticar o fio sobre os planos do cubo e a cada volta completa do mesmo avançar o suporte 1 ou 2 mm, sempre mantendo uma tensão específica para o diâmetro do fio utilizado.

Ao final obtemos quatro planos de fios paralelos que são passados para matrizes menores e depois soldados na placa de circuito impresso.

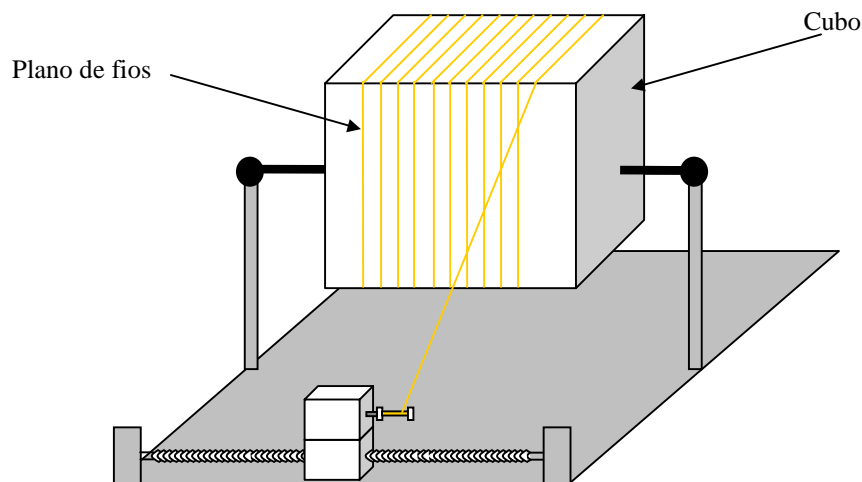


Figura 4: Estrutura cúbica giratória para tecelagem do plano de fios

Este procedimento era feito manualmente gerando um grande desgaste físico ao operador, um longo tempo de execução (cerca de 2 horas) e havia a necessidade de guias nas arestas do cubo para manter o fio no espaçamento desejado.

## **II - Automação do Processo de Tecelagem de Eletrodos Multifilares**

A necessidade de automação resultou em um projeto de mecatrônica onde um programa executado em um microcomputador envia sinais pela porta paralela para um circuito eletrônico que controla motores e sensores que fazem a mecânica funcionar da forma desejada. A tarefa de operação consiste simplesmente em posicionar o fio no suporte e soldar sua ponta em uma das arestas do cubo. Feito isso, basta acionar um botão na tela do microcomputador e o processo de tecelagem ocorre com precisão, confiabilidade e segurança. Isso reduziu o tempo de tecelagem para menos de 5 minutos.

### **2.1 - Sistema Mecânico**

Para automatizar este sistema foram adaptados motores, sensores, polias, correias e rolamentos de forma a obter sincronismo durante processo. Utilizamos 3 motores:

- Dois motores de passo (um para movimento do cubo e outro para o movimento do fuso).
- Um motor DC (para controle de tensão do fio).

Para controlar e proteger o sistema e permitir um movimento sincronizado foram incorporados sensores ópticos e sensores mecânicos:

- Dois sensores ópticos : um para sinalizar o rompimento do fio (figura 5) e outro para contagem de voltas do cubo (figura 8).
- Dois sensores mecânicos para detectar fim de curso (figura 6)

### **2.2 - Monitoramento e Alinhamento do Fio**

Um sistema de polias foi projetado para guiar e manter o alinhamento do fio, que é monitorado por um sensor óptico ativado em caso de rompimento. A bobina do fio foi acoplada a um motor DC, de

modo que o fio é mantido sob tensão durante todo o processo de tecelagem. A disposição do motor DC é esboçada na figura 6.

No conjunto de polias mostrado na figura 5 pode-se observar que uma delas é acoplada a um pêndulo, e fica suspensa durante o processo de tecelagem. Se o fio se romper a polia é solta fazendo com que o pêndulo passe pelo sensor óptico, que quando ativo envia um sinal ao computador que interrompe o movimento dos motores.

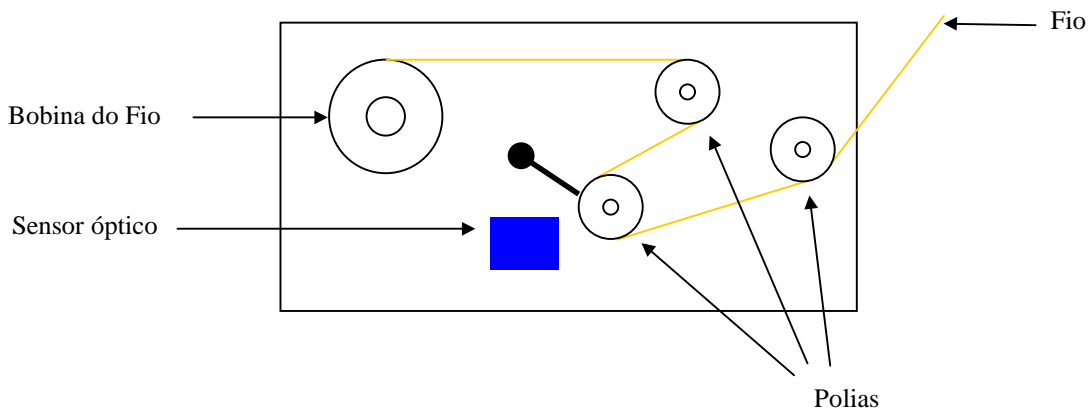


Figura 5: Conjunto de polias para guiar o fio e sensor óptico para rompimento.

### 2.3 - Proteção de Fim de Curso

Os sensores mecânicos de fim de curso funcionam como proteção para o caso de o suporte chegar a uma das extremidades do fuso. Quando isso acontece é enviado um sinal ao computador, que interrompe o movimento dos motores. Estes sensores foram fixados no suporte conforme é mostrado na figura 6.

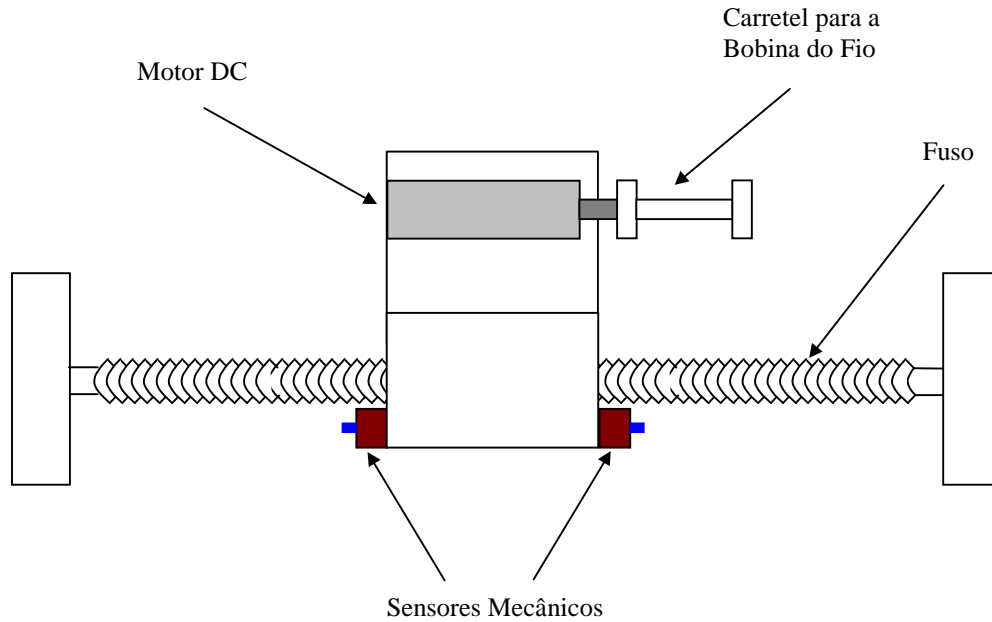


Figura 6: Suporte mecânico com sensores de fim de curso e fuso.

## 2.4 - Movimento do Cubo

Um motor de passo foi acoplado à lateral de uma das hastes que sustentam o cubo. Através de uma correia e duas polias sincronizadas este motor faz a estrutura cúbica girar (figura 7). O sensor óptico de contagem foi fixado na mesma haste e é acionado por um pino adaptado a uma das arestas do cubo, de modo que a cada giro completo o sensor é ativado, enviando um sinal para o microcomputador fazer um outro motor avançar o suporte em um número de passos programados (figura 8).



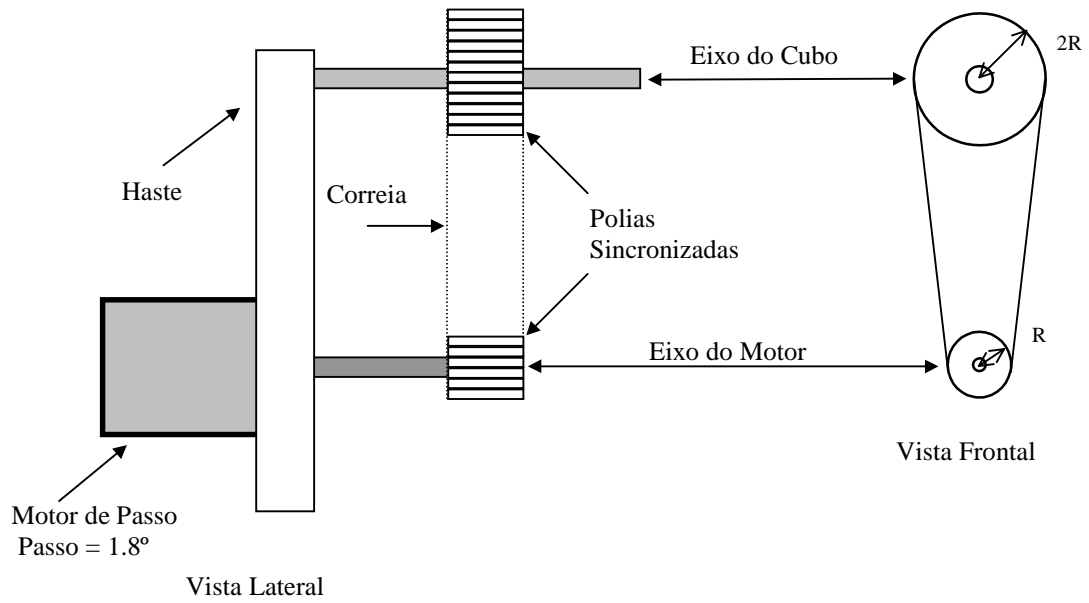


Figura 7: Vistas lateral e frontal do acoplamento entre o eixo do cubo e o motor.

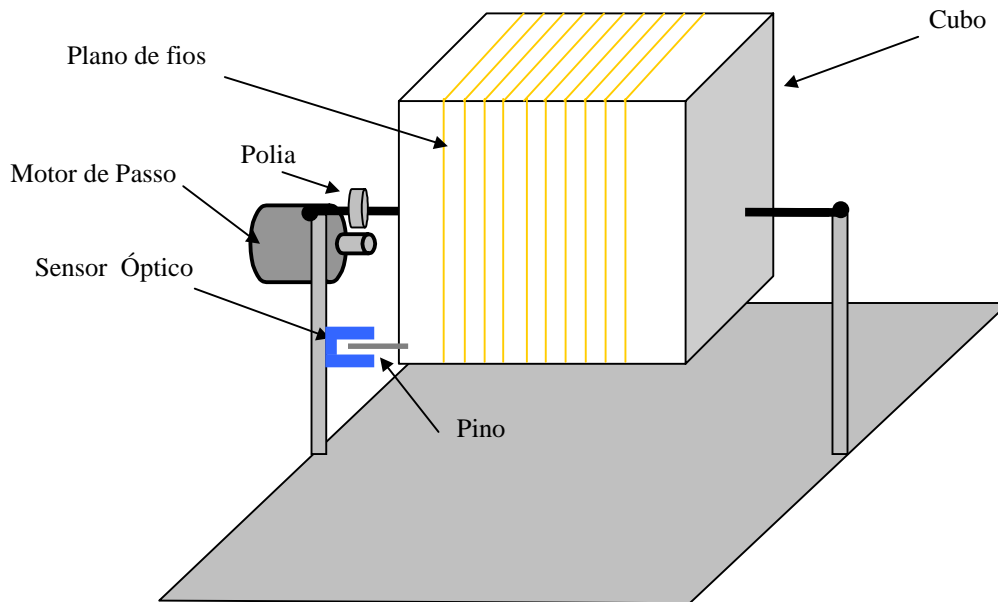


Figura 8: Sensor óptico de contagem e pino de acionamento

## 2.5 - Movimento do Suporte do Fio

Para movimentar o suporte do fio foi utilizado um motor de passo com o eixo acoplado diretamente ao fuso (figura9).

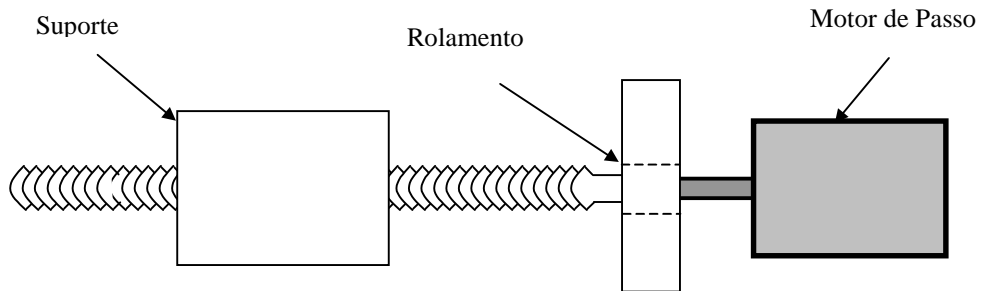


Figura 9: Motor de passo acoplado ao fuso

### 2.5.1 - Especificações

O fuso utilizado tem passo de 1mm ( $\Delta X = 1\text{mm}$ ), para avançar este valor o motor precisa dar um giro completo ( $\Delta \theta = 360^\circ$ ). Para isso é necessário avançar 200 passos ou 400 1/2 passos conforme tabela abaixo.

	1/2 Passo	Passo
<b><math>\Delta \theta</math> Motor</b>	0.9°	1.8°
<b><math>\Delta X</math> FUSO</b>	2.5 $\mu\text{m}$	5.0 $\mu\text{m}$
<b># Passos / mm</b>	400	200

### III - Circuito Eletrônico

Para controlar o sistema foi desenvolvido um circuito eletrônico que controla o avanço dos dois motores de passo, controla o torque do motor DC e polariza os sensores.

#### 3.1 - Controle de Motor de Passo Bipolar

O controle de motores de passo bipolares é obtido com um circuito integrado específico para esta aplicação - CI L297 - que gera as fases para o movimento do motor de acordo com os sinais TTL aplicados as suas entradas de controle. Para gerar a corrente necessária para o funcionamento do motor as fases são aplicadas à entrada de um driver de potência – CI L298 – cuja saída é acoplada diretamente aos enrolamentos do motor, utilizando diodos de partida rápida para proteger o circuito da força-contratraz gerada no induzido. A montagem é feita conforme mostrado na figura 10.

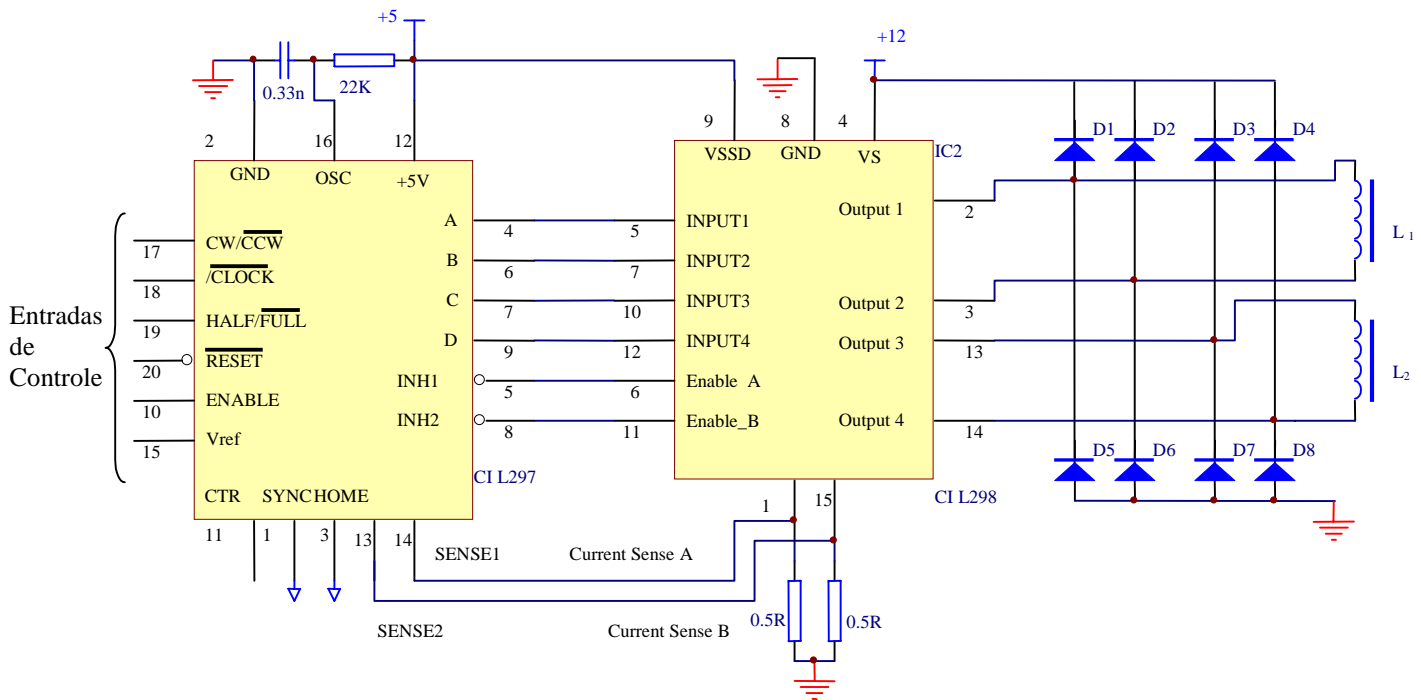


Figura 10: Circuito Eletrônico para controle de um motor de passo bipolar.

### 3.1.1 - Especificações de Funcionamento

O circuito mostrado na figura 10 controla apenas um motor de passo bipolar. O CI L297 recebe os sinais de controle pela porta paralela do microcomputador e gera as fases para acionamento do motor. A tensão de referência –  $V_{ref}$  – ( pino 15 ), define a corrente de pico máxima,  $I_L$ , na bobina do motor. Utilizamos a relação:

$$I_L = V_{ref} / R_{sense}.$$

\*  $V_{ref}$  pode variar de 0 a 3 volts.

\*  $R_{sense}$  – pinos 13 e 14.

Os sinais de controle utilizados no CI L297 foram:

$\overline{\text{CLOCK}}$  – Pino 18 - Controla o avanço do motor. Ativo em nível baixo, cada borda de subida do sinal corresponde a um passo do motor. Logo quanto maior a frequência do pulso de clock maior o número de transições e maior a velocidade.

$\overline{\text{CW/CCW}}$  – Pino 17 – Define o sentido do giro do Motor. ( horário ou anti-horário)

$\overline{\text{HALF/FULL}}$  – Pino 19 – Permite optar pelo avanço em passo ( $1.8^\circ$ ) ou meio passo ( $0.9^\circ$ ). O que aumenta a resolução do controle.

$\overline{\text{ENABLE}}$  – Pino 10 - Estando desativada, ou seja, em nível baixo, todas as saídas A, B ,C, D, INH1 e INH2 são levadas para zero e o motor fica desenergizado. Isto ocorre mesmo havendo pulso de  $\overline{\text{CLOCK}}$ . Estando ativada, em nível alto, o motor fica energizado mesmo que o pino  $\overline{\text{RESET}}$  esteja ativado ou que não haja pulso de  $\overline{\text{CLOCK}}$ . Neste caso o motor fica parado, porem, tensionado pela indução nas bobinas.

### 3.2 - Controle de Corrente para Motor DC

Para controlar o torque no motor DC, que mantêm a tensão mecânica do fio, foi utilizado o circuito da figura 11. Como se pode notar, à medida em que se varia a resistência do *trimpot* a corrente de coletor  $I_L$  do transistor aumenta ou diminui, controlando assim a corrente no motor DC.

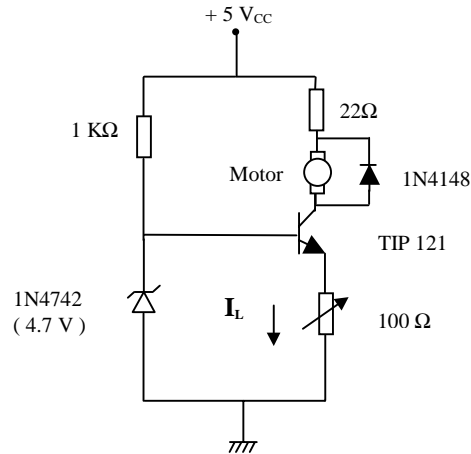


Figura 11: Circuito controlador de corrente para motor DC.

### 3.3 - Circuito Gerador de Pulsos

Para gerar o pulso de CLOCK para o CI L297 que controla o motor do cubo, foi montado um circuito gerador de onda quadrada. Com o amplificador operacional CI555 em uma configuração típica de multivibrador astável, um sinal é gerado no pino 3 do CI e, como se mostra na figura 12, cuja frequência é alterada de acordo com a variação do *trimpot*.

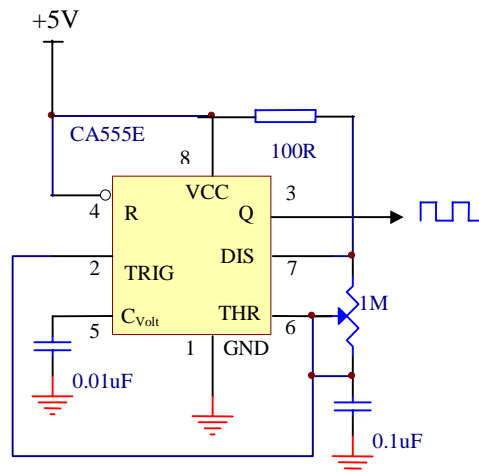


Figura 12: Gerador de Pulsos

### 3.4 - Polarização dos Sensores

Os quatro sensores foram polarizados em nível lógico "1" e quando ativos levam suas saídas para nível lógico "0", as quais estão ligadas ao endereço de entrada da porta paralela.

Os dois sensores mecânicos são chaves normalmente abertas (figura13).

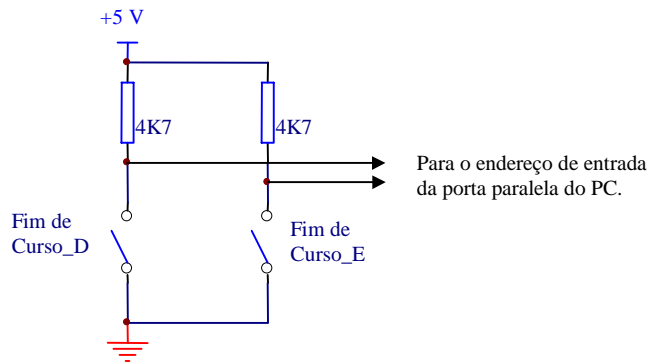


Figura 13: Sensores Mecânicos de fim de curso – Ativos em nível baixo.

Os sensores ópticos (figura 14) foram polarizados de forma a se obter um nível lógico "1" no emissor do fototransistor enquanto este estiver recebendo radiação do *led* infravermelho. Quando esta incidência é interrompida o transistor entra em corte e o sinal no emissor cai para nível lógico "0".

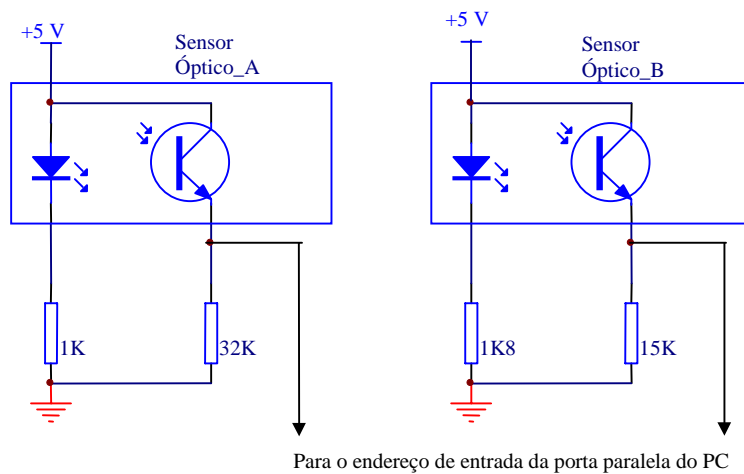


Figura 14: Sensores Ópticos (contador e monitor do fio) – Ativos em nível lógico "0".

## IV - Interface Paralela

### 4.1 - A Porta Paralela do PC

A Porta Paralela do PC é uma interface de microcomputador para conexões externas que possui três endereços distintos: dois endereços para Instruções ou dados de Saída e um para a Instruções ou dados de Entrada. Utilizando qualquer linguagem de programação pode-se acessar estes endereços para enviar e receber dados de qualquer periférico conectado à porta. A conexão é feita através de um conector do tipo DB25 e a relação entre a pinagem no conector, endereço e sinal relacionado está descrita nas tabelas a seguir.

#### Endereço 378H – Instrução de Saída

Referência no Barramento de Dados do Microprocessador	D7	D6	D5	D4	D3	D2	D1	D0
Sinal Relacionado no Conector	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0
Pino no Conector DB25	9	8	7	6	5	4	3	2

Por exemplo: enviando o dado 237d (ou 11101101b, ou EDh) para este endereço, teremos na saída do conector a configuração mostrada na tabela abaixo.

Referência no Barramento de Dados do Microprocessador	D7	D6	D5	D4	D3	D2	D1	D0
<b>Sinal no Conector</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>
Pino no Conector DB25	9	8	7	6	5	4	3	2

**Endereço 37AH – Instrução de Saída**

Referência na Barra de Dados do	D7	D6	D5	D4	D3	D2	D1	D0
Microprocessador								
Sinal Relacionado no Conector	-	-	-	-	$\overline{C3}$	C2	$\overline{C1}$	$\overline{C0}$
Pino no Conector DB25	-	-	-	-	17	16	14	1

Obs.: Observe que os sinais C3, C1 E C0 são enviados invertidos. Por exemplo, se o dado XXXX1011b for enviado para este endereço será lido na saída do conector o dado XXXX0000b.

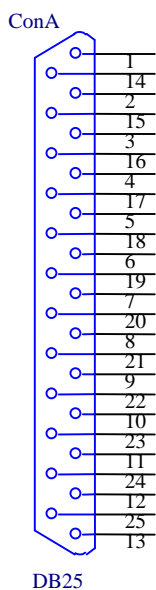
**Endereço 379H – Instrução de Entrada**

Referência na Barra de Dados do	D7	D6	D5	D4	D3	D2	D1	D0
Microprocessador								
Sinal Relacionado no Conector	$\overline{E7}$	E6	E5	E4	E3	-	-	-
Pino no Conector DB25	11	10	12	13	15	-	-	-

Obs: O sinal aplicado neste endereço tem o seu bit mais significativo (E7) lido de forma invertida pelo microcomputador.



A conexão com a porta paralela ocorre através de um conector do tipo DB25. A configuração de sinais apresentada abaixo é utilizada para interface com impressora.



Pino	Nome	Descrição
1	/STROBE	Strobe
2	D0	Data Bit 0
3	D1	Data Bit 1
4	D2	Data Bit 2
5	D3	Data Bit 3
6	D4	Data Bit 4
7	D5	Data Bit 5
8	D6	Data Bit 6
9	D7	Data Bit 7
10	/ACK	Acknowledge
11	BUSY	Busy
12	PE	Paper End
13	SELIN	Select In
14	/AUTOFD	Autofeed
15	/ERROR	Error
16	/INIT	Initialize
17	/SEL	Select
18 - 25	GND	Signal Ground

#### 4.2 - Configuração da Porta Paralela na Automação do Sistema

O endereço de Instrução de Saída **378h** foi escolhido para enviar os dados de controle para os dois motores. O esquema elétrico do circuito foi montado de modo que os primeiros 4 bits deste endereço vão para as entradas do CI L297 que controla o motor do cubo giratório e os 4 bits seguintes vão para as entradas do outro CI L297 que controla o motor do fuso. As entradas utilizadas são as que

definem sentido, passo/meio-passo, habilitação e desabilitação do motor e estão ligadas conforme a seguinte tabela.

### **Tabela de Dados de Controle - Endereço 378h**

Referência na Barra de Dados do Microprocessador	D7	D6	D5	D4	D3	D2	D1	D0
Sinal Relacionado no Conector	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0
Pino no Conector DB25	9	8	7	6	5	4	3	2
<b>Sinal de Controle para 2 CI's L297</b>	$\overline{\text{RESET}}$	ENABLE	$\overline{\text{HALF/}}\overline{\text{FULL}}\overline{\text{STEP}}$	$\overline{\text{CW/}}\overline{\text{CCW}}$	$\overline{\text{RESET}}$	ENABLE	$\overline{\text{HALF/}}\overline{\text{FULL}}\overline{\text{STEP}}$	$\overline{\text{CW/}}\overline{\text{CCW}}$

Controlo do Motor do Fuso

Controlo do Motor do Cubo

O controle da velocidade do motor do fuso é feito enviando ao endereço **37Ah** um dado que alterna-se entre "1" e "0". Essa transição gera um pulso, que é enviado a o pino de clock do CI L297, onde, o tempo de permanência em cada valor determina a frequência deste sinal.

O endereço **379h** de instrução de entrada foi configurado para ler os sensores ópticos e mecânicos e estão ligados conforme a tabela abaixo.

### **Endereço 379H – Instrução de Entrada**

Referência na Barra de Dados do Microprocessador	D7	D6	D5	D4	D3	D2	D1	D0
Sinal Relacionado no Conector	$\overline{\text{E7}}$	E6	E5	E4	E3	-	-	-
Pino no Conector DB25	11	10	12	13	15	-	-	-
<b>Sensores</b>	<b>Contador</b>	-	<b>Monitora o Fio</b>	<b>Fim de Curso Esquerdo</b>	<b>Fim de Curso Direito</b>	-	-	-

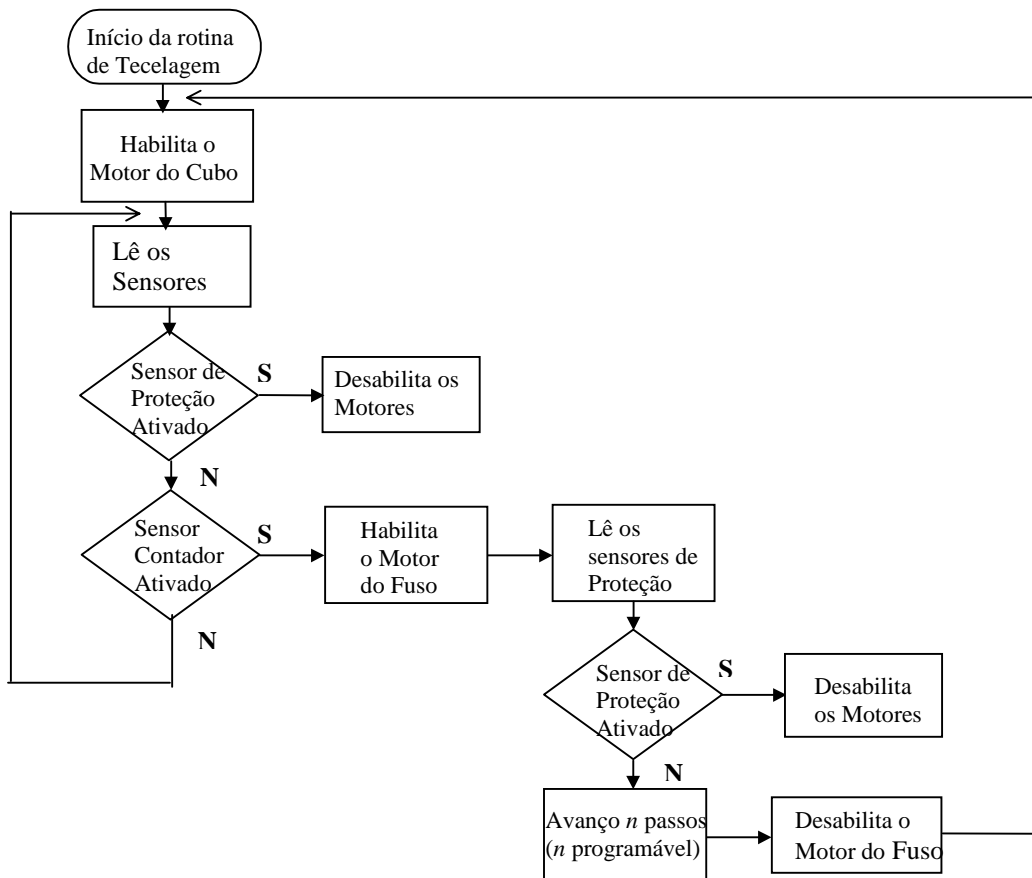
## V - Programação

O ambiente de programação utilizado para acessar a porta paralela e enviar e ler os sinais de controle do sistema foi o Delphi.

O programa permite um controle individual de cada motor no que se refere ao sentido, ao avanço em passo ou meio-passo, e o controle da velocidade do motor do fuso, e possui uma tecla “*Iniciar Tecelagem*” que coloca em sincronismo o movimento dos motores e a leitura dos sensores.

### 5.1 - Rotina “*Iniciar Tecelagem*”

A tecla “*Iniciar Tecelagem*” faz com que o processo de tecelagem inicialize, colocando os motores e os sensores em funcionamento sincronizado com a possibilidade de definir o avanço do suporte do fio para 1 ou 2 mm. Durante este processo, todos os sensores de proteção estão sendo monitorados. No caso de acionamento de algum deles, o programa envia sinais para desabilitar os dois motores e o sistema é parado. O funcionamento da rotina de tecelagem do programa de automação é esquematizado no fluxograma abaixo.





### 5.3 - Código Fonte do Programa

```
var
  FormAutom: TFormAutom;
  Clock : Boolean = False;
  Start : Boolean = False;
  EnableCubo, EnableFuso, PassoCubo, PassoFuso, DirCubo, DirFuso, Port : Byte;

implementation

{$R *.DFM}

procedure PortWriteByte (Addr : Word; Value : Byte); assembler; register;
asm
  XCHG AX,DX
  OUT DX,AL
end;

function PortReadByte (Addr : Word) : Byte; assembler; register;
asm
  MOV DX,AX
  IN AL,DX
end;

procedure TFormAutom.ButtonStartClick(Sender: TObject);
var
  X : Integer;
begin
  ButtonStop.Enabled := True;
  ButtonStart.Enabled := False;
  ButtonAuto.Enabled := False;
  RadioButtonLeftClick (Sender);
```

```
//Port := 128 + EnableFuso * 64 + PassoFuso * 32 + DirFuso * 16 + 8 + EnableCubo * 4 + PassoCubo * 2 +
DirCubo;
//PortWriteByte (888, Port);
Clock := True;
while Clock do
begin
  case PortReadByte (889) or 71 of
    95, 111, 119, 223, 239, 247 : ButtonStopClick (Sender);
  end;

  Application.ProcessMessages;

  if CheckBoxFuso.Checked then
  begin
    for X := 1 to SpinEditClockFuso.Value do PortWriteByte (890, 0);
    for X := 1 to spineditclockFuso.Value do PortWriteByte (890, 1);
  end;
end;

end;

procedure TFormAutom.ButtonStopClick(Sender: TObject);
begin
  ButtonStop.Enabled := False;
  ButtonStart.Enabled := True;
  ButtonStopAuto.Enabled := False;
  ButtonAuto.Enabled := True;
  GroupBoxSentido.Enabled := True;
  GroupBoxStep.Enabled := True;
  GroupBoxSentidoFuso.Enabled := True;
  GroupBoxStepFuso.Enabled := True;
  CheckBoxPlano.Enabled := True;
  CheckBoxFuso.Enabled := True;

  PortWriteByte (888, 0);
```

```
PortWriteByte (890, 0);
```

```
Clock := False;
```

```
Start := False;
```

```
end;
```

```
procedure TFormAutom.RadioButtonLeftClick(Sender: TObject);
```

```
begin
```

```
if RadioButtonFull.Checked then PassoCubo := 0 else PassoCubo := 1;
```

```
if RadioButtonLeft.Checked then DirCubo := 0 else DirCubo := 1;
```

```
if RadioButtonFullFuso.Checked then PassoFuso := 0 else PassoFuso := 1;
```

```
if RadioButtonLeftFuso.Checked then DirFuso := 0 else DirFuso := 1;
```

```
if CheckBoxPlano.Checked and not ButtonStart.Enabled then EnableCubo := 1 else EnableCubo := 0;
```

```
if CheckBoxFuso.Checked then EnableFuso := 1 else EnableFuso := 0;
```

```
Port := 128 + EnableFuso * 64 + PassoFuso * 32 + DirFuso * 16 + 8 + EnableCubo * 4 + PassoCubo * 2 +  
DirCubo;
```

```
PortWriteByte (888, Port);
```

```
end;
```

```
procedure TFormAutom.ButtonTestClick(Sender: TObject);
```

```
begin
```

```
LabelTest.Caption := 'Dado: ' + IntToStr (PortReadByte (StrToInt (EditTest.Text)) or 71)
```

```
end;
```

```
procedure TFormAutom.ButtonAutoClick(Sender: TObject);
```

```
var
```

```
Auto : Boolean;
```

```
j : Integer;
```

```
i, Contador : Word;
```

```
begin
```

```
ButtonStopAuto.Enabled := True;
```

```
ButtonAuto.Enabled := False;
```

```
ButtonStart.Enabled := False;
```

```
GroupBoxSentido.Enabled := False;
GroupBoxStep.Enabled := False;
GroupBoxSentidoFuso.Enabled := False;
GroupBoxStepFuso.Enabled := False;
CheckBoxPlano.Enabled := False;
CheckBoxFuso.Enabled := False;

Contador := 0;

Start := True;
while Start do
begin
    PortWriteByte (888, 14); //Faz o motor da Cubo girar

    Application.ProcessMessages;
    //Loop prende o programa até o pino passar pelo sensor de contagem
    Auto := True;
    while Auto do
    begin
        Application.ProcessMessages;
        case PortReadByte (889) or 71 of
            95, 111, 119, 223, 239, 247 : ButtonStopClick (Sender);
        end;

        if not Start then Exit;

        case PortReadByte (889) or 71 of 255 : Auto := False end;
    end;

    Application.ProcessMessages;
    PortWriteByte (888, 222); // Habilita o motor do Fuso e do Cubo

    //Loop para o Motor do Fuso avançar 1 ou 2 mm
    for i := 1 to 133 * (ComboBoxStep.ItemIndex + 1) do
```



```
begin
  for j := 1 to SpinEditClockFuso.Value do PortWriteByte (890, 1);
  case PortReadByte (889) or 71 of
    95, 111, 119, 223, 239, 247 : ButtonStopClick (Sender);
  end;

  if not Start then Exit;

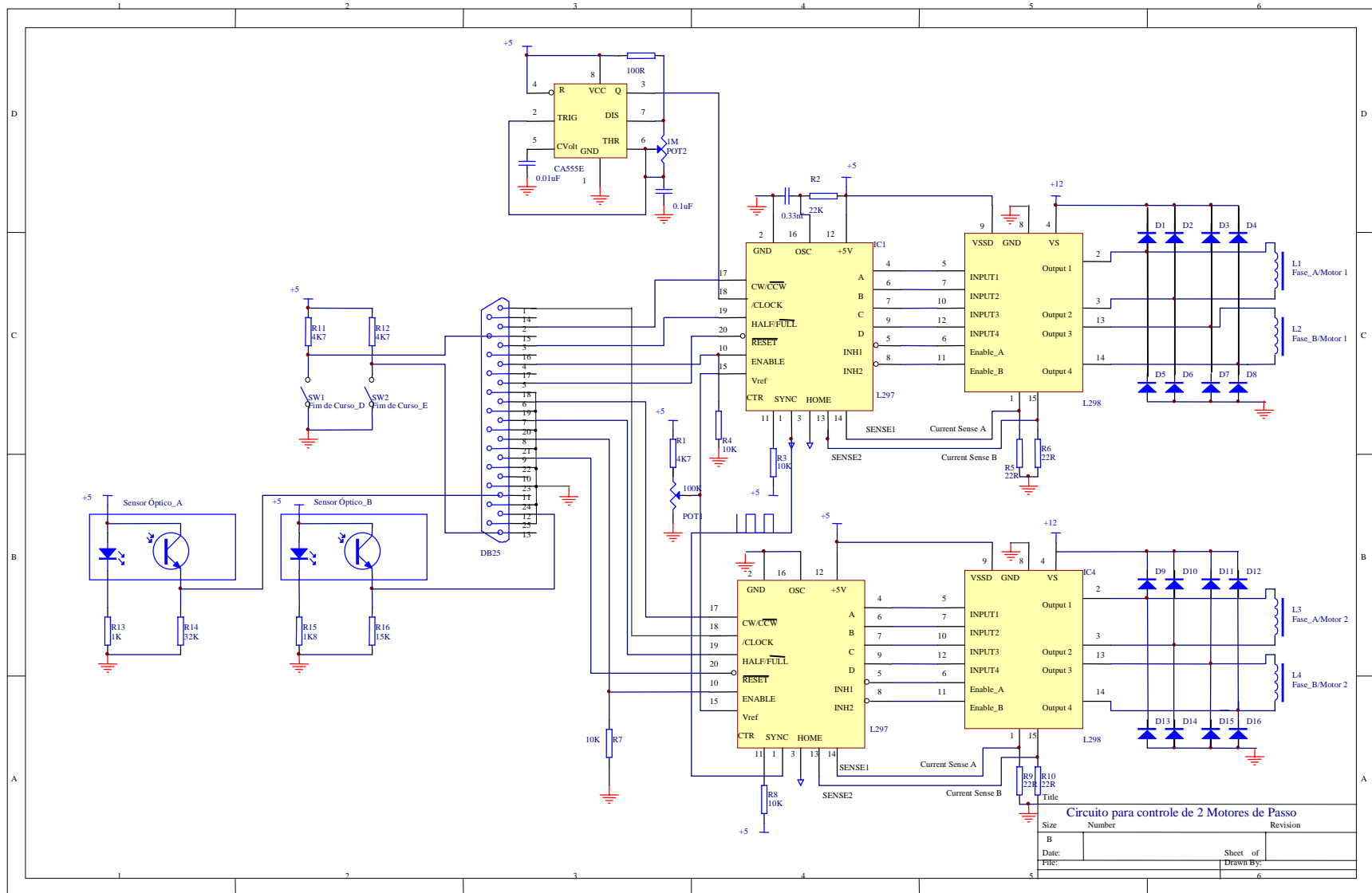
  for j := 1 to SpinEditClockFuso.Value do PortWriteByte (890, 0);

  Inc (Contador);
  EditContador.Text := IntToStr (Contador);
end;
end;
end;

procedure TFormAutom.FormCreate(Sender: TObject);
begin
  ComboBoxStep.ItemIndex := 0;
end;

end.
```

ANEXO A - Esquema Eléctrico



Circuito para controle de 2 Motores de Passo

Size	Number	Revision
B		
Date:		Sheet of
File:		Drawn By:
		6

## ANEXO B - Diagrama Esquemático

