

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO
CAMPUS SÃO JOSÉ DOS CAMPOS

Augusto Magno

Matheus Alves

Pedro Vinícius

Mesa de Posicionamento de Precisão

Trabalho de Conclusão de Curso
apresentado ao Instituto Federal de
Educação, Ciência e Tecnologia de São
Paulo – Campus São José dos Campos,
como requisito para obtenção do Título de
Técnico em Automação Industrial sob
orientação do Professor Cláudio Luís dos
Santos.

São José dos Campos

2014

BANCA EXAMINADORA

Trabalho de Conclusão de Curso (TCC) defendido e aprovado em
16 de junho de 2014, pela banca examinadora constituída pelos professores:

Prof. MsC Cláudio Luís dos Santos
Orientador

Prof. Dr. Getúlio de Vasconcelos

Aos nossos queridos pais,
Professores e amigos,
Com muito carinho,

Dedico

Agradecimentos

Ao orientador, Claudio Luís dos Santos, por acreditar em nossa capacidade, valorizar o nosso trabalho e contribuir para o nosso crescimento pessoal e intelectual.

Ao técnico de laboratório, Danilo Braga, pelo auxílio, companhia e disposição.

A professora, Vania Battestin Wiendl, pela contribuição, paciência e amizade.

Aos professores Aguinaldo Cardozo da Costa Filho, André Luis Mendes Moura, Amita Muralikrishna, Cláudio Luís dos Santos, Irineu dos Santos Yassuda, João Sinohara da Silva Souza, Lineu Alves Lima Filho, Luís Carlos Pires Videira, Luís Gustavo de Oliveira, Ricardo Becker, Edson Vinci por toda a disposição, atenção e ensinamentos dispensados no decorrer desses anos.

Aos nossos pais, Mauriza e Lauro, Patrícia e Erivaldo, Darci e Maria pelo incentivo, companheirismo e amor.

Sumário

LISTA DE TABELAS	vi
LISTA DE FIGURAS	vii
LISTA DE ABREVIATURAS E SÍMBOLOS	viii
RESUMO	ix
ABSTRACT	x
1. INTRODUÇÃO.....	1
2. OBJETIVO	2
3. REVISÃO BIBLIOGRÁFICA	2
3.1. Motor de passo	2
3.2. Cabo DB25 e a Porta Paralela (LPT)	2
3.3. O software de Programação Visual Basic 6.....	5
4. MATERIAIS E MÉTODOS	5
4.1. Programação	6
4.2. Circuito Eletrônico	7
4.3. Fonte de Tensão	7
4.4. Placa Controladora	9
4.5. Motores.....	12
4.6. Características Elétricas.....	13
5. RESULTADOS E DISCUSSÕES	13
6. CONCLUSÃO.....	14
7. REFERÊNCIAS BIBLIOGRÁFICAS	15
8. ANEXOS	16
8.1. Código Fonte	16

LISTA DE TABELAS

Tabela 1: Endereços da porta paralela.....	4
Tabela 2: Características elétricas.....	13
Tabela 3: Valor da velocidade conforme intervalo de tempo entre os passos..	14

LISTA DE FIGURAS

Figura 1: Ilustração do Conector Macho do Cabo Paralelo.....	3
Figura 2: Conector Macho do Cabo Paralelo.....	3
Figura 3: Design do programa em visual basic 6.....	6
Figura 4: Esquema elétrico da fonte de alimentação.....	8
Figura 5: Esquema da parte posterior da fonte.....	8
Figura 6: Esquema elétrico da placa controladora.....	10
Figura 7: Ligação do conector centronics à placa.....	11
Figura 8: Vista frontal do equipamento.....	12
Figura 9: Ligação do conector ao motor.....	12

LISTA DE ABREVIATURAS E SÍMBOLOS

CNC	Controle Numérico Computacional
V	Volts (Unidade de tensão Elétrica)
PC	Personal Computer
IBM	International Business Machines
SPP	Standard Parallel Port
CPU	Central processing unit
BUS	Barramento
EPP	Enhanced Parallel Port
ECP	Enhanced Capabilities Port
DMA	direct memory access
FIFO	First In, First Out

RESUMO

Neste trabalho foi realizado o desenvolvimento de um programa para acionamento de uma mesa CNC de posicionamento de precisão. Esta mesa consiste a partir de coordenadas usadas com o propósito de proporcionar uma referência ao programador durante o desenvolvimento do programa. O sistema de acionamento da mesa deste TCC é constituído de: motor de passo, programa de acionamento desenvolvido no Visual Basic 6, cabo DB25, porta paralela, microprocessador e computador. O programa de acionamento possibilitou uma precisão de deslocamento da mesa da ordem de 0,015 mm. Além do controle do deslocamento, o programa também possibilita o controle da velocidade de deslocamento da mesa na faixa de 0,076 a 0,750 mm/s.

Palavras-chave: Deslocamento, Posicionamento, Programação, mesa CNC.

ABSTRACT

This work was conducted to develop a program to drive a CNC precision positioning table. A positioning table is used from the purpose of providing a reference to the programmer during the development of the program coordinates. The precision positioning table system was developed using: a stepper motor, a microprocessor, software developed in Visual Basic 6.0 to control the movements, the parallel port LPT and a computer, with the control software was possible to control the displacement of the table with a high accurately. Besides the displacement, with the program was also possible to control the speed of the movement in the range of 0,076 to 0,750 mm/s.

Keywords: Displacement, Positioning, Programming, CNC table.

1. INTRODUÇÃO

Uma mesa de posicionamento consiste a partir de coordenadas usadas como propósito de proporcionar uma referência ao operador durante o desenvolvimento dos seus projetos. Indústrias como, por exemplo, de manufatura, onde se inclui o ramo automobilístico, entre outros é estritamente importante trabalhar com eixos e esteiras de alta precisão. Considerando uma linha de montagem, com equipamentos para executar inúmeras tarefas ao longo da linha, é necessário que a matéria prima a ser trabalhada seja posicionada com exatidão.

A mesa consiste de dois eixos cartesianos, que podem ser ajustados de acordo com a necessidade do operador, com a possibilidade de escolha de sentido e velocidade de deslocamento dos eixos. Essa mesa apresenta semelhanças às funcionalidades empregadas em uma máquina CNC (Controle Numérico Computadorizado), porém apresentando um diferencial, onde a peça usinada é feita com o pó metálico, sendo fundido camada a camada, desse modo a perda de material em uma usinagem é bem reduzida, uma vez que o pó metálico que não foi fundido a peça pode ser reutilizado na usinagem de outra peça posteriormente. O projeto foi desenvolvido com base em outro já existente, de uma mesa de sinterização, que através da energia gerada por um laser, funde o pó de uma liga metálica específica.

2. OBJETIVO

Construir e programar uma mesa de posicionamento de precisão que consiste em movimentar carga no eixo escolhido, afim de melhor se trabalhar os produtos, tendo em vista a precisão e eficiência do posicionamento da carga que está sendo movimentada.

3. REVISÃO BIBLIOGRÁFICA

3.1. Motor de passo

Motores de Passo são dispositivos eletromecânicos que convertem pulsos elétricos em movimentos mecânicos que geram variações angulares discretas. O rotor ou eixo de um motor de passo é rotacionado em pequenos incrementos angulares, denominados “passos”, quando pulsos elétricos são aplicados em uma determinada sequência nos terminais deste. A rotação de tais motores é diretamente relacionada aos impulsos elétricos que são recebidos, bem como a sequência a qual tais pulsos são aplicados, reflete diretamente na direção a qual o motor gira. A velocidade que o rotor gira, é dada pela frequência de pulsos recebidos e o valor do ângulo correspondente à rotação causada, está diretamente relacionado com o número de pulsos aplicados (SCRIBD).

3.2. Cabo DB25 e a Porta Paralela (LPT)

O DB25 é um conector que fica na parte de trás do gabinete é através deste, que cabo paralelo se conecta ao computador para poder enviar e receber dados. No DB25, um pino está em nível lógico 0 quando a tensão elétrica no mesmo está entre 0 à 0,4V. Um pino se encontra em nível lógico 1 quando a tensão elétrica no mesmo está acima de 3,1V e até 5V (ROGERCOM).

A Figura 1 e a Figura 2 mostram o conector padrão DB25, com 25 pinos, onde cada pino tem um nome com a sua identificação (ROGERCOM).

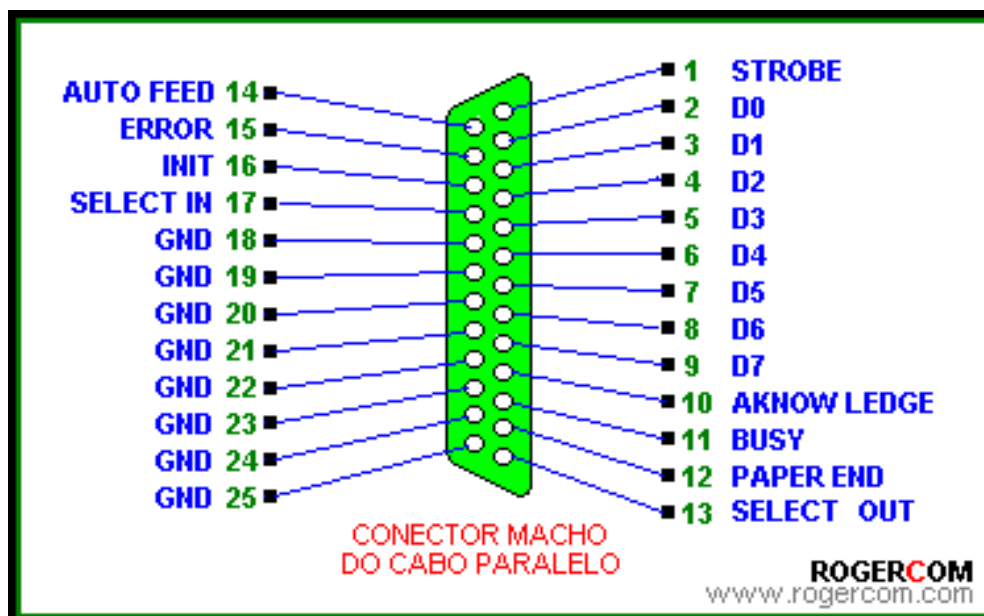


Figura 1: Ilustração do Conector Macho do Cabo Paralelo (Fonte: ROGERCOM).



Figura 2: Conector Macho do Cabo Paralelo (Fonte: ROGERCOM).

Pode ser observado na Figura 1 que os pinos 2 a 9 referem-se às saídas digitais da porta paralela. Estas saídas podem ser utilizadas para acionamento de drivers de motores de passo. Para isso, torna-se necessário o desenvolvimento de um aplicativo que pode ser feito, por exemplo, em visual basic, para controlar a direção, assim como velocidade do motor.

A porta paralela é uma interface de comunicação entre o computador e um periférico. Quando a IBM criou seu primeiro PC ou Computador Pessoal, a ideia era conectar a essa Porta uma impressora, mas atualmente, são vários os periféricos que se utilizam desta Porta para enviar e receber dados para o computador

(exemplos: Scanners, Câmeras de vídeo, Unidade de disco removível e outros) (ROGERCOM).

Existem diferentes modelos de porta paralela, os quais são: a) Transmissão unidirecional: A porta paralela SPP pode chegar a uma taxa de transmissão de dados a 150KB/s. Comunica-se com a CPU utilizando um BUS de dados de 8 bits. Para a transmissão de dados entre periféricos são usado 4 bits por vez (ROGERCOM). b) Transmissão bidirecional: A porta avançada EPP chega a atingir uma taxa de transferência de 2 MB/s. Para atingir essa velocidade, será necessário um cabo especial. Comunica-se com a CPU utilizando um BUS de dados de 32 bits. Para a transmissão de dados entre periféricos são usado 8 bits por vez. A porta avançada ECP tem as mesmas características que a EPP, porém, utiliza DMA, sem a necessidade do uso do processador, para a transferência de dados. Utiliza também um buffer FIFO de 16 bytes (ROGERCOM).

As Portas Paralelas são nomeadas pelo computador como LPTs. Exemplo: LPT1, LPT2, LPT3, etc... A Porta física padrão do computador é a LPT1, e seus endereços são: 378h (para enviar um byte de dados pela Porta), 378+1h (para receber um valor através da Porta) e, 378+2h (para enviar dados). Às vezes pode está disponível a LPT2, e seus endereços são: 278h, 278+1h e 278+2h, com as mesmas funções dos endereços da porta LPT1 respectivamente. A TABELA 1 mostra os endereços da porta paralela (ROGERCOM).

Tabela 1: Endereços da porta paralela (ROGERCOM).

Nome da Porta	Endereço de memória	Endereço da Porta		Descrição
LPT1	0000:0408	378 hexadecimal	888 decimal	Endereço base
LPT2	0000:040A	378 hexadecimal	632decimal	Endereço base

3.3. O software de Programação Visual Basic 6

O software Microsoft Visual Basic é um pacote para desenvolvimento de aplicações visuais para ambiente Windows baseado na linguagem de programação Basic. É orientado a eventos, o que quer dizer que trata ocorrências que dão início a alguma rotina de trabalho: o programa fica parado até que algo aconteça. Quer dizer também que ele permite o uso de objetos, mas não a sua criação, pois não é uma linguagem orientada a objetos. Objetos são estruturas que combinam propriedades e métodos (UTFPR).

As propriedades são características dos objetos, que podem ser acessadas e/ou alteradas pelo programador tanto em tempo de projeto (quando o projeto está sendo desenvolvido) quanto em tempo de execução (quando o aplicativo está sendo executado). Já os métodos são rotinas internas ao objeto que servem para executar determinadas ações. Para exemplificar, pense em uma bicicleta azul. A cor azul é uma característica da bicicleta, ou uma propriedade dela. Já um método seria o ato de pedalar, que é a rotina necessária para fazer a bicicleta andar. Para programação em VB, usamos uma versão da linguagem Basic estruturada para a construção de procedimentos e funções que podem estar associados aos eventos dos objetos de sua aplicação. O VB também faz uma verificação automática de sintaxe dos comandos, e possui recursos avançados de compilação e rastreamento de erros (UTFPR).

4. MATERIAIS E MÉTODOS

Para desenvolvimento da mesa de posicionamento, foram necessários os seguintes itens:

- Motor de passo;
- Software Microsoft Visual Studio;
- Microsoft Visual Basic 6;
- Cabo DB25;
- Porta paralela;

- Microprocessador;
- Computador.

Além dos itens de hardware, foi necessário também o desenvolvimento de um aplicativo para controle do posicionamento da mesa. A Figura 3 mostra o aplicativo desenvolvido em VB6 para controle da mesa.

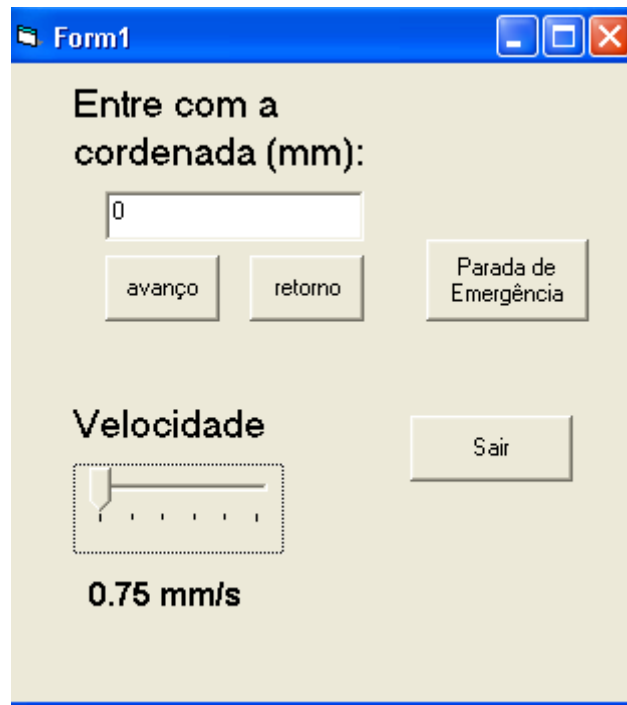


Figura 3: Design do programa em visual basic 6.

Neste aplicativo, foram desenvolvidos os seguintes componentes: 1 *trackbar* para controle de velocidade da mesa, dois *buttons* para direção (subir ou descer), 1 *button* para parada de emergência e 1 *text* para que o usuário informe qual a distância para que a mesa seja movimentada, para apresentar o título como: “Velocidade”, “Entre com a coordenada (mm)” e o indicador de velocidade atual foram feitas com *Labels*.

4.1. Programação

A atividade de programação em relação a parte de interface gráfica foi realizada utilizando o software Microsoft Visual Studio 2010, onde foi montado a parte visual da utilização funcional do programa. A partir do programa desenvolvido

em Visual Studio 2010, foi desenvolvido o programa em VB6 devido à necessidade de utilização da porta paralela para acionamento do microprocessador.

4.2. Circuito Eletrônico

Este equipamento foi projetado para controlar os movimentos de dois motores de passo controlados via software, o circuito eletrônico é composto de uma fonte de alimentação de 12 VDC e uma placa controladora de alto desempenho que comanda os movimentos dos motores de passo. A placa controladora suporta uma tensão de 12 VDC a 35 VDC, e permite que a corrente elétrica fornecida para os motores de passo seja ajustada de 0 a 1 A, eliminando variações indesejáveis na corrente elétrica responsável pelo mau funcionamento dos motores de passo (Ferraz, Lima, Vasconcelos).

Os controles do motor de passo, como a velocidade e o sentido de rotação, são gerados através de um software que envia pulsos lógicos (1 ou 0) para a placa controladora (Ferraz, Lima, Vasconcelos).

4.3. Fonte de Tensão

Constitui-se por uma fonte de tensão cuja saída é de 12 VDC e 1A, possui alimentação bi-voltagem (110/220 V), onde o ajuste é feito por uma chave posicionada na parte traseira do equipamento (Ferraz, Lima, Vasconcelos).

A Figura 4 apresenta o esquema elétrico, e a Figura 5 apresenta o desenho esquemático da parte posterior da fonte de alimentação (Ferraz, Lima, Vasconcelos).

4.4. Placa Controladora

A placa controladora utilizada no dispositivo de sinterização é de alto desempenho, cujo esquema é apresentado na Figura 6. Esta placa controla os movimentos dos motores de passo, cuja alimentação é fornecida pela fonte de alimentação (Ferraz, Lima, Vasconcelos).

A placa controlada possui vários acopladores ópticos, cuja função é de isolar os sinais lógicos enviados pelo computador através da porta paralela, impedindo assim que problemas elétricos possam causar danos ao computador (Ferraz, Lima, Vasconcelos).

O software envia sinal para 8 – bits CMOS Microcontroller, onde é feita a leitura de movimento do motor, se será meio passo (HALF) ou passo completo (FULL). Em seguida o sinal é dividido em 6 sinais. Sendo que 3 sinais seguem para um Full Bridge Driver que controla uma bobina do motor de passo e os outros 3 são enviados para outro Full Bridge Driver que controla uma outra bobina, controlando assim, 1 motor de passo. Para o controle do outro motor de passo o procedimento é o mesmo, porém, em CIs diferentes (Ferraz, Lima, Vasconcelos).

O Full Bridge Driver, além de controlar o motor de passo, também é capaz de controlar a corrente elétrica que pode ser ajustada entre 0 a 1 A de acordo com o valor da corrente do motor de passo (Ferraz, Lima, Vasconcelos).

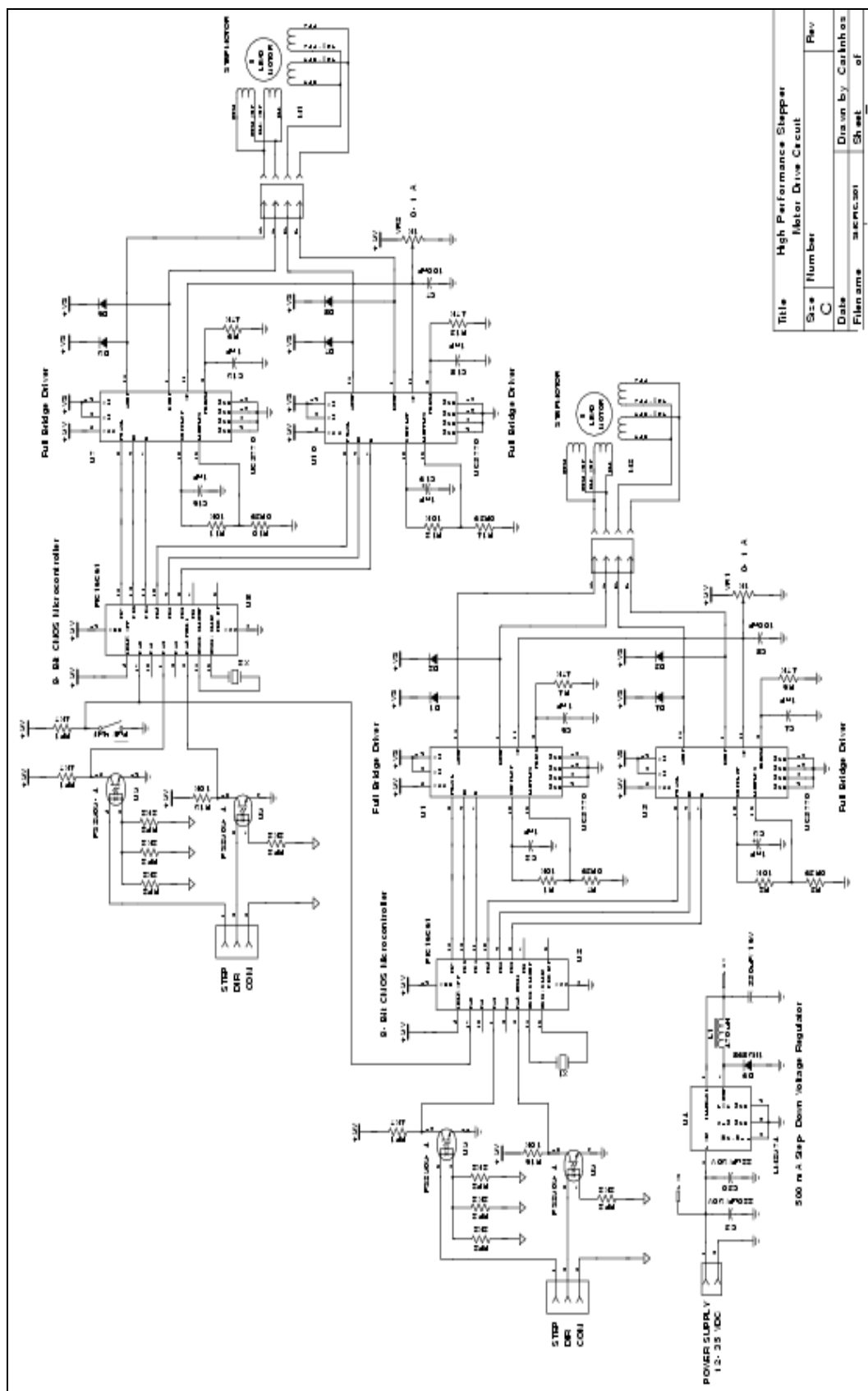


Figura 6: Esquema elétrico da placa controladora (Ferraz, Lima, Vasconcelos).

A ligação do conector centronics à placa é apresentada na Figura 7, *placa onde S1 e D1 correspondem ao motor 1 e o S2 e D2 correspondem ao motor 2:*

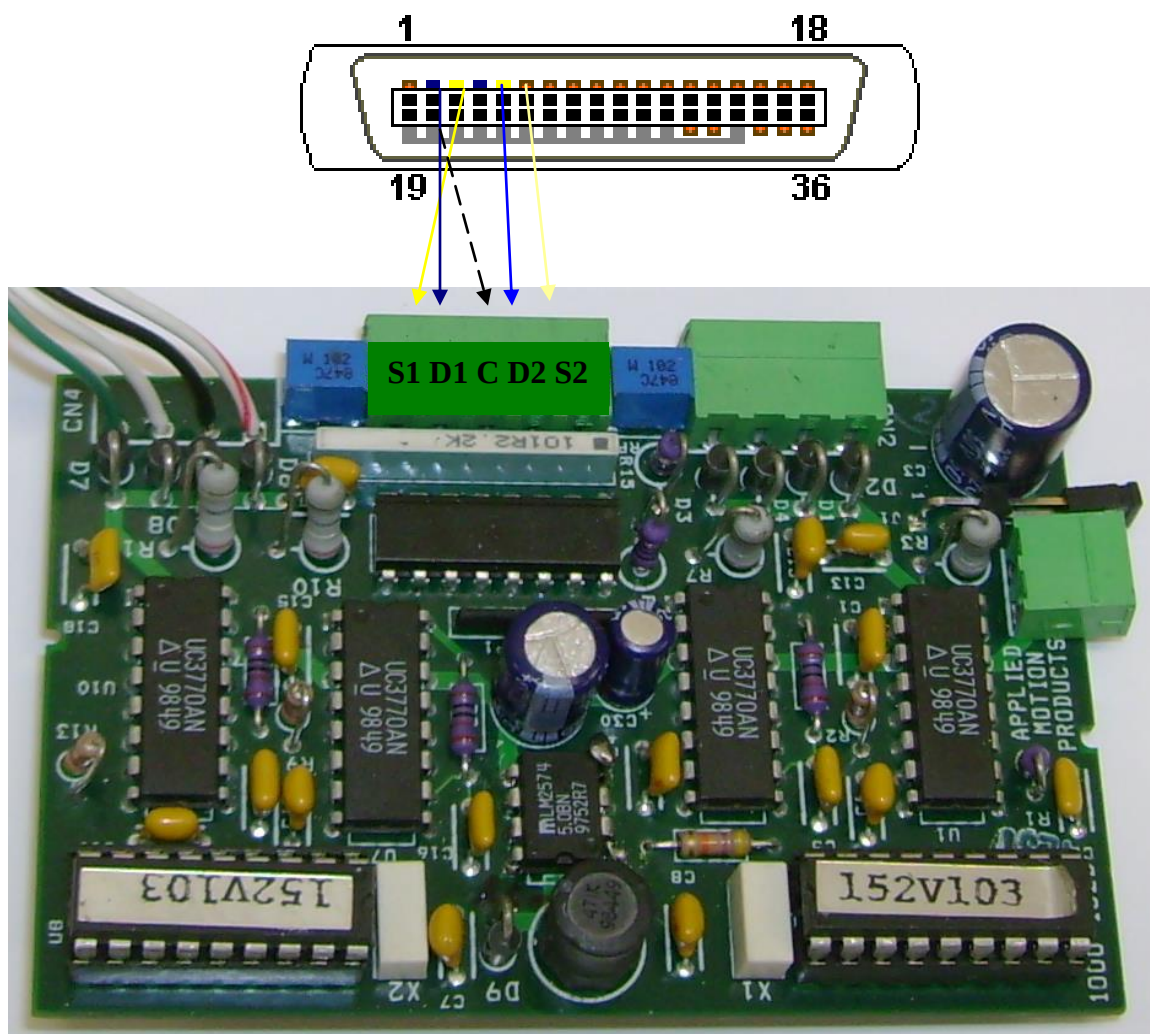


Figura 7: Ligação do conector centronics à (Ferraz, Lima, Vasconcelos).

No cabo centronics o pino 2 e 3 são responsáveis pelo controle do motor 1, onde o pino 2 está ligado ao step responsável pela entrada do pulso e o pino 3 responsável pela direção do motor. O mesmo procedimento é válido para o pino 4 que é ligado ao step e o pino 5 ligado à direção, que será responsável pelo controle do motor 2, os pinos de 19 à 30 foram ligados em um único ponto, onde o mesmo foi ligado ao pino 33 formando o comum ou terra (Ferraz, Lima, Vasconcelos).

4.5. Motores

São constituídos por dois motores de passo, modelo 23 D – 6102 C, 5.1 V, 1A, com passos de 0,9° e 1,8°. Cada motor de passo é ligado a um conector e a ordem dos conectores é apresentada na Figura 8. As ligações de cada motor ao conector foram feitas conforme a Figura 9 (Ferraz, Lima, Vasconcelos).

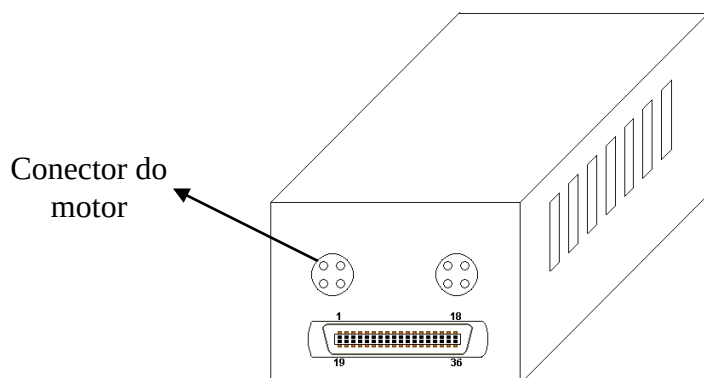


Figura 8: Vista frontal do equipamento (Ferraz, Lima, Vasconcelos).

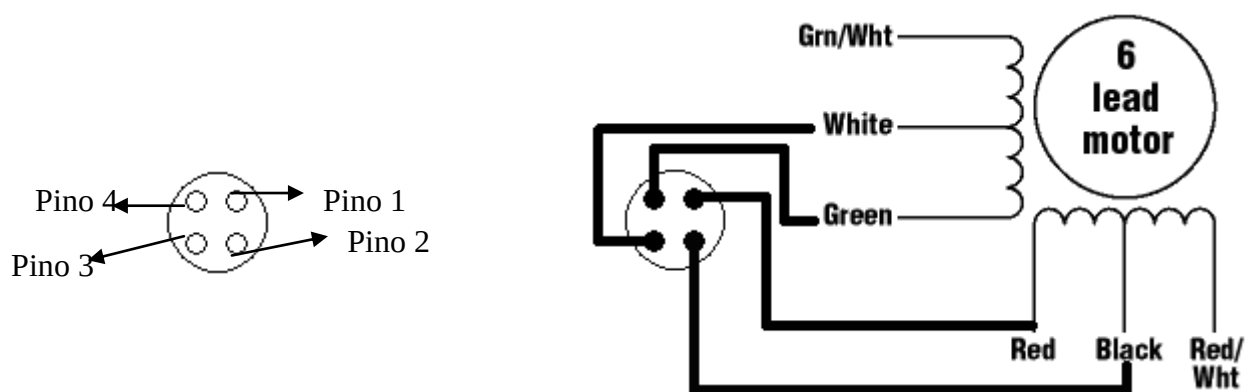


Figura 9: Ligação do conector ao motor (Ferraz, Lima, Vasconcelos).

4.6. Características Elétricas

As características elétricas dos componentes utilizados podem ser observadas na Tabela 2.

Tabela 2: Características elétricas (Ferraz, Lima, Vasconcelos).

	<i>Tensão de Entrada</i>	<i>Tensão de Saída</i>	<i>Corrente</i>
<i>Fonte de Tensão</i>	110/220 V	15 V	~1A
<i>Placa Controladora</i>	15 a 35 V	5 V	0 a 1A
<i>Motor de Passo</i>	5V	-----	0 a 1A

5. RESULTADOS E DISCUSSÕES

Inicialmente o trabalho foi realizado no software Microsoft Visual Studio 2010, instalado em um computador com Windows 7, que continha um driver de bloqueio de comunicação com as portas serial e paralela, o que poderia ser resolvido instalando um outro driver de saída, o outboard. Porém, como o computador utilizado pertencia ao IFSP - SJC e ainda estava dentro do prazo de garantia, não substituímos peças do mesmo. Devido a esse fato, foi necessária a mudança para um computador com o Windows XP e a utilização do software Visual Basic 6.0. Isto possibilitou o acesso à porta paralela e, conseqüentemente, o acionamento do microprocessador para controle da mesa.

Foram realizados vários testes, sendo necessárias mudanças nas ligações do motor, para que fosse possível o giro do mesmo, tanto no sentido horário, como no sentido anti-horário. Depois de algum tempo, foi admissível que o motor tivesse rotação bidirecional, porém ainda sem conseguir com que o mesmo parasse no meio de seu ciclo, ou seja, no caso de uma emergência. Por fim, algumas linhas de programa foram alteradas para que através do programa de acionamento fosse possível parar o motor sem esperar o ciclo total de movimento.

Outro ponto importante neste trabalho é que foi atingido o objetivo de precisão, ou seja, o eixo mover-se 15µm a cada pulso dado no motor de passo. Além da precisão no movimento, o programa também possibilitou o controle de velocidade, conforme mostrado na Tabela 3.

Tabela 3: Valor da velocidade conforme intervalo de tempo entre os passos.

Δt (ms)	Velocidade (mm/s)
150.6	0.076
120.6	0.096
90.6	0.130
60.6	0.190
30.6	0.380
0.6	0.750

Pode ser observado na Tabela 3 que o controle do tempo entre passos possibilita o controle da velocidade de deslocamento.

6. CONCLUSÃO

Apenas com a utilização do Windows XP, foi possível a realização do projeto nos parâmetros previamente estabelecidos, atingindo o objetivo em precisão, utilização de materiais e tempo disponível para a execução. O controle do deslocamento e da velocidade possibilita uma mesa de posicionamento de alta precisão, mesmo com carga no eixo do motor. O intervalo de controle de velocidade da mesa é pequeno (0.076 a 0.750 mm/s), contudo dentro deste intervalo ela realiza o deslocamento com alta precisão.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Projeto IFM-CNPq. Prototipagem rápida: fabricação com laser. Getúlio de Vasconcelos, Clayson Carvalho Ferraz e Milton S. Fernandes de Lima.
- [2] ROGERCOM. <http://www.rogercom.com/pparalela/introducao.htm#conecto>. Acesso em: 10/03/2014.
- [3] SCRIBD. <http://pt.scribd.com/doc/22751145/ConceitosMotordePasso>>. Motor de passo. Acesso em 17/03/2014.
- [4] UTFPR. <http://pessoal.utfpr.edu.br/gustavo/apostila%20de%20vb.pdf>. Microsoft Visual Basic 6. Acessado em: 15/10/2013
- [5] VICTORTRUCCO. <http://www.victortrucco.com/CNC/MotoresdePasso/MotoresdePasso.asp> Acesso em: 10/03/2014.

8. ANEXOS

8.1. Código Fonte

```
Private Sub Command1_Click()
```

```
    'Pistão descida'
```

```
        Out 888, 11  'Definição do endereço da porta-paralela a ser  
utilizada'
```

```
        Sleep (z)    'Atribuindo variável z a Sleep para intervalo de tempo'
```

```
        j = 1
```

```
    'Laço de Repetição'
```

```
        While j <= x * 83
```

```
            Out 888, 1
```

```
                Sleep (z)
```

```
DoEvents
```

```
    j = j + 1
```

```
    y = y + 1
```

```
Out 888, 0
```

```
    a = Inp(889)
```

```
Out 888, 0
```

```
    If a <> 28 Then
```

```
        Label4.Caption = " sensor não ativado "
```

```
    Else
```

```
        Label4.Caption = "sensor ativado"
```

```
End If
```

```
    Wend
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    'Pistão Subida'
```

```

        j = 1

        While j <= 83 * x

            Out 888, 7
            Sleep (z)

            Out 888, 5

            DoEvents
            j = j + 1
        Wend
End Sub

Private Sub Command5_Click()

    Out 888, 0

    If z <> 0 Then
        j = 1

        While j <= x * 83.33333333333333

            Out 888, 4
            Sleep (z)
            Out 888, 5
            DoEvents
            j = j + 1

        Wend
    End If
End Sub

Private Sub Command6_Click()
    End
End Sub

Private Sub Form_Load()

End Sub

```

```

Private Sub Slider1_Click()
    Out 888, 256

    'Atribuindo valor de z para os vários pontos de Slider'
    If Slider1.Value = 0 Then
        z = 150.6
        Label3.Caption = "0.076 mm/s"
    ElseIf Slider1.Value = 1 Then
        z = 120.6
        Label3.Caption = "0.096 mm/s"
    ElseIf Slider1.Value = 2 Then
        z = 90.6
        Label3.Caption = "0.13 mm/s"
    ElseIf Slider1.Value = 3 Then
        z = 60.6
        Label3.Caption = "0.19 mm/s"
    ElseIf Slider1.Value = 4 Then
        z = 30.6
        Label3.Caption = "0.38 mm/s"
    ElseIf Slider1.Value = 5 Then
        z = 0.6
        Label3.Caption = "0.75 mm/s"
    End If
End Sub

Private Sub Text1_Change()
    'Atribuindo valores às variáveis, a partir da text1
    x = Val(Text1.Text)
    y = Val(Text1.Text)
End Sub

```