

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

GUILHERME YUKISHIGUE NISHIZAWA

Impressora de circuitos usando Visão Computacional

São Carlos
2010

GUILHERME YUKISHIGUE NISHIZAWA

IMPRESSORA DE CIRCUITOS USANDO VISÃO COMPUTACIONAL

Trabalho de Conclusão de Curso apresentado
à Escola de Engenharia de São Carlos, da
Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase em
Eletrônica

ORIENTADOR: Professor Doutor Adilson Gonzaga

São Carlos
2010

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

N724i Nishizawa, Guilherme Yukishigue
Impressora de circuitos usando visão
computacional / Guilherme Yukishigue Nishizawa ;
orientador: Adilson Gonzaga. -- São Carlos, 2010.

Trabalho de Conclusão de Curso (Graduação em
Engenharia Elétrica com ênfase em Eletrônica) -- Escola
de Engenharia de São Carlos da Universidade de São
Paulo, 2010.

1. Fresagem. 2. Visão computacional. 3. Visual basic.
4. Motores de passo. I. Título.

Dedico esse trabalho e a graduação para os meus pais e amigos que ajudaram a vencer mais essa etapa da minha vida.

Agradecimentos

Agradeço a todos meus familiares que me dão força e motivação para sempre seguir em frente e sobrepor todas as barreiras que apareceram ao longo desses anos. A minha turma que trouxe grandes aprendizados tanto pra carreira profissional quanto pra vida. Em especial agradeço ao Rodrigo Neves que serviu de mestre ao longo desses anos, ao Samir e Mateus por todo o divertimento fornecido, ao Gabriel, ele sabe o motivo, além de todos os outros grandes amigos que espero levar pra vida, Biondo, Natália, Lari, Zeitona e a Débora Auschwitz. Ao Professor Adílson pela orientação e atenção fornecida durante a graduação e o trabalho.

E a todos que contribuíram de alguma forma para a conclusão deste trabalho.

“Todas estas questões, devidamente ponderadas, levantam dúvidas sobre se a complexidade dos estudos efetuados afeta positivamente a correta previsão das direções preferenciais no sentido do progresso. Por outro lado, o acompanhamento das preferências de consumo estende o alcance e a importância das diversas correntes de pensamento.” (Marcus Túlio Cícero)

Resumo

Este trabalho propõe um projeto completo de uma fresa CNC (Controle Numérico Computadorizado), desde a entrada de dados por meio de um microcomputador pessoal (PC) até o hardware final. O software escrito em linguagem Visual Basic envolve conceitos de Visão Computacional introduzidos durante a graduação. Assim, é possível obter uma interface gráfica amigável e resultados que atendam devidamente às necessidades do sistema, isto é, processar uma imagem de circuito impresso de forma a detectar suas bordas e transformar essas informações em instruções para a fresa. O hardware foi projetado de forma a possibilitar uma impressão em dois eixos impulsionados por motores de passo que se movimentam segundo instruções transmitidas através da porta paralela do PC e geradas a partir do processamento de imagem realizado pelo software. O intuito principal é encontrar uma forma simples e acessível de suprir as necessidades do público técnico amador de obter um resultado com maior qualidade e comodidade quando se trata de placas para montagens elétrico/eletrônica.

Palavras-Chave: Fresagem, Visão Computacional, Visual Basic, motores de passo

Abstract

This work come up with a full project of a CNC Router, since the input of data that is made through an user interface from the computer to the final ending hardware. The software has been developed in Visual Basic Language and cover concepts of Computer Vision introduced during the graduation, making possible to obtain a friendly graphic user interface and results that attend properly to the needs of the system, in other words, to process a circuit image to detect its boards and transform those informations in instructions to the drilling. The hardware is designed in a way to enable an impression in two axes impulsed by step motors that move according to the instructions transmitted through the parallel port and are generated from the software processing. The main aim is to find a simple and acessible way to supply the needs of the technical unprofessional public of obtaining a result with mode quality and confort when talking about boards for electrical/electronical composition.

Key Words: Router, Computer Vision, Visual Basic, step motors

Lista de Figuras

Figura 1 - Kit de Confecção de Circuitos.....	21
Figura 2 - Diagrama de Blocos do Processo de Fresagem.....	22
Figura 3 - Fresa IEC3020.....	24
Figura 4 - Fresa SHG0404.....	25
Figura 5 - Fresa LPKF ProtoMat S100.....	26
Figura 6 - Ambiente de Desenvolvimento Visual Basic 6.0.....	27
Figura 7 - Exemplo de Arquivos Gerados pelo Visual Basic 6.....	29
Figura 8 - Protótipo da Fresa desenvolvida.....	31
Figura 9 - Interface do Programa da Fresa.....	33
Figura 10 - Diagrama de Fluxo de Dados do Programa.....	34
Figura 11 - Cursor de Movimentação da Fresa.....	35
Figura 12 - Antes e Depois da Detecção das Bordas.....	35
Figura 13 - Painel de Controle do Processo.....	36
Figura 14 - Processo de Fresagem em Tempo Real.....	36
Figura 15 - Trecho do Código na Tela do Compilador.....	37
Figura 16 - Resultado do Algoritmo de Detecção de Bordas.....	41
Figura 17- Porta Paralela.....	41
Figura 18 - Conector DB25 da Porta Paralela no modo SPP.....	42
Figura 19 - Equivalente eletrônico da Fresa.....	43
Figura 20 - Imagem utilizada para Determinação de Velocidade de Deslocamento.....	44
Figura 21 - Triângulo para Medição da Potência.....	45
Figura 23- Caneta Utilizada para os Testes de Resolução.....	47
Figura 24 - Foto do procedimento final utilizando a caneta.....	47
Figura 25 - Imagem Original.....	51
Figura 26 - Sobel e Threshold.....	51
Figura 27 - Prewitt e Threshold.....	51
Figura 28 - Roberts e Threshold.....	51
Figura 29 - Canny e Threshold.....	51
Figura 30 - Algoritmo desenvolvido para fresa.....	51
Figura 31 - Resultado do Filtro de Roberts e o Threshold.....	51

Figura 32 - Resultado do algoritmo desenvolvido.....	51
Figura 33- Imagem Coloria Original	52
Figura 34 - Roberts e Threshold	52
Figura 35 - Algoritmo da Fresa	52
Figura 36 - Filtro de Roberts para Circuitos Eletrônicos	52
Figura 37- Retângulo para inferir a potência.....	53

Lista de Tabelas

Tabela 1 - Especificações Técnicas da Fresa IEC3020.....	24
Tabela 2 – Especificações Técnicas da Fresa SHG0404	25
Tabela 3 - Especificações Técnicas da Fresa LPKF ProtoMat S100.....	26
Tabela 4 - Tipos de Dados da Linguagem Visual Basic	28
Tabela 5- Tipos de Arquivos Utilizados pelo Visual Basic 6	29
Tabela 6 - Descrição da Base do Hardware	32
Tabela 7 - Descrição da Plataforma do Hardware	32
Tabela 8 - Descrição do Suporte do Cabeçote Fresador do Hardware	32
Tabela 9 - Requisitos para o Software da Fresa	33
Tabela 10 - Velocidade de Deslocamento da Fresa (1)	48
Tabela 11 - Velocidade de Deslocamento da Fresa (2)	48
Tabela 12 - Velocidade de Deslocamento da Fresa (3)	49
Tabela 13 - Área útil de impressão	49
Tabela 14 - Comparativo das Tecnologias	50
Tabela 15 - Comparação dos Filtros de Detecção de Bordas para uma Imagem Qualquer .	51
Tabela 16 - Comparação do Filtro de Roberts com o algoritmo desenvolvido para um esquema de circuito.....	51
Tabela 17 - Comparação dos Filtros de Detecção de Bordas para uma Imagem Colorida..	52
Tabela 18 - Resultados do trecho 1.....	53
Tabela 19 - Resultados do trecho 2.....	53
Tabela 20 - Resultados do trecho 3.....	53

Sumário

Agradecimentos	7
Resumo	11
Abstract	13
Lista de Figuras.....	15
Lista de Tabelas	17
Sumário	19
Capítulo 1 – Introdução.....	21
1.1 Apresentação	21
1.2 Objetivos	21
1.3 Organização da Monografia.....	22
Capítulo 2 – Revisão de Literatura	23
2.1 Fresa CNC.....	23
2.2 Produtos Existentes no Mercado	23
2.2.1 Fresadora IEC3020.	23
2.2.2 Fresadora SHG0404 CNC Router	24
2.2.3 Fresadora LPKF ProtoMat S100	26
2.3 Conceitos de Linguagem de Software e Algoritmos Utilizados	27
2.3.1 Linguagem Visual Basic	27
2.3.2 Conceitos de Visão Computacional.....	29
2.4 Considerações Finais	30
Capítulo 3 – Materiais e Métodos.....	31
3.1 Materiais Utilizados.....	31
3.1.1 Peça de Hardware	31
3.1.2 Aplicativo Desenvolvido.....	33
3.1.3 Codificação.....	37
3.1.4 Algoritmo de Detecção das Bordas	40
3.1.5 Circuito de Potência da Impressora	41
3.2 Metodologia	43

3.2.1	Testes de Limites do Sistema Projetado	43
3.2.2	Comparativo do Algoritmo de Detecção de Bordas	44
3.2.3	Consumo de Energia	45
Capítulo 4 – Resultados e Conclusões		47
4.1	Testes dos limites do sistema projetado.....	47
4.1.1	Velocidade de Deslocamento da fresa	47
4.1.2	Área útil de impressão	49
4.1.3	Resolução de impressão	49
4.1.4	Conclusões e comparativo com outras tecnologias	49
4.2	Comparativo do algoritmo de detecção de bordas	50
4.2.1	Resultados	50
4.2.2	Conclusões sobre o algoritmo	52
4.3	Consumo de energia	53
4.4	Conclusões Gerais	54
4.5	Trabalhos Futuros.....	54
Referências Bibliográficas		56
Anexo 1 – Linguagem Matlab		57
Apêndice A – Código do arquivo Paralela.bas.....		59
Apêndice B – Código do arquivo Variáveis.bas.....		63
Apêndice C – Código do arquivo fresadora.frm.....		65
Apêndice D – Pseudo-Código para entendimento do algoritmo principal		77

Capítulo 1 – Introdução

1.1 Apresentação

Hoje em dia o processo de confecção de circuitos elétricos por parte de amadores é necessariamente feita a mão através de canetas de tinta permanente (Figura 1). Outras técnicas podem ser encontradas, como por exemplo, impressões utilizando impressoras a laser, utilizando máquinas de cópias. Todas elas envolvem o processo de corrosão por Percloroeto de Ferro (FeCl_3) que possui alto poder oxidante e por isso é capaz de corroer as partes que não entram em contato com a substância (PATNAIK, 2007). Independente da forma como é feita a marcação na placa de cobre, todos esses processos demandam algum trabalho por parte do usuário e tempo para que ocorra o processo químico. Unindo conceitos adquiridos ao longo do curso de graduação é possível apontar uma solução melhor para esse problema que acaba servindo como motivação e um desafio a ser superado.



Figura 1 - Kit de Confecção de Circuitos

1.2 Objetivos

Este trabalho propõe como objetivo principal o desenvolvimento de um projeto de baixo custo, para usuários que necessitam produzir placas para circuitos impressos.

O projeto contempla o desenvolvimento de um sistema completo (hardware/software) para a prototipagem de circuitos impressos tendo como entrada uma imagem BITMAP. Após o posicionamento inicial da fresa e o carregamento da imagem, esta é armazenada em forma de matriz que será processada através de um algoritmo de detecção de bordas. Essas bordas são convertidas em instruções para os motores do hardware que movimentarão a fresa. O fluxo de operação do projeto proposto, pode ser visto no diagrama de blocos da Figura 2.

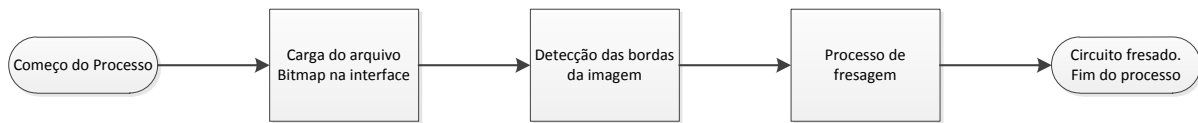


Figura 2 - Diagrama de Blocos do Processo de Fresagem

É importante ressaltar o caráter multidisciplinar do projeto, uma vez que conceitos de mecânica, eletrônica, eletrônica de potência, computação, sistemas digitais e visão computacional devem ser aplicados para o sucesso do projeto. A integração de todos esses fatores de forma bem sucedida, se torna, portanto, um dos objetivos do projeto.

Com este intuito, o trabalho envolve conhecimentos adquiridos de vários campos abordados durante o curso de graduação (eletro-mecânica, eletrônica, programação, visão computacional) para gerar uma solução que atenda a esta necessidade.

1.3 Organização da Monografia

A monografia se estrutura de acordo com a segmentação em capítulos proposta a seguir:

Capítulo 1: Introdução - Faz uma sinopse do tema da monografia de modo a fornecer uma idéia abrangente do projeto.

Capítulo 2: Revisão de Literatura – Tem como objetivo apresentar o tema (Fresas CNC), bem como as tecnologias atuais que se assemelham a tecnologia do projeto. Outro ponto importante e diferencial desse projeto é a parte de visão computacional, que também será citada nesse capítulo.

Capítulo 3: Materiais e Métodos – Este capítulo apresenta como o trabalho foi desenvolvido, cobrindo a metodologia e os materiais utilizados para que os objetivos fossem alcançados.

Capítulo 4: Resultados e Conclusões – Este capítulo apresenta as análises e os resultados obtidos com a implementação da metodologia. Aponta também possíveis melhorias a serem feitas futuramente e a conclusão do projeto.

Capítulo 2 – Revisão de Literatura

2.1 Fresa CNC

Em resumo uma fresadora é uma máquina de movimento contínuo destinada à usinagem de materiais (SÉRGIO, 1967). A fresadora CNC (Computer Numeric Control ou em Português Controle Numérico Computadorizado) funciona com um controlador numérico que permite o controle de máquinas e é utilizado principalmente em centros de usinagem. Permite o controle simultâneo de vários eixos, através de uma lista de movimentos escrita num código específico (código G ou G-code). Na década de 1940 foi desenvolvido o NC (Controle Numérico) que evoluiu posteriormente para o CNC. A utilização de CNC permite a produção de peças complexas com grande precisão, especialmente quando associado a programas de CAD/CAM.

A introdução do CNC na indústria mudou radicalmente os processos industriais. Curvas são facilmente cortadas, complexas estruturas com 3 dimensões tornam-se relativamente fáceis de produzir e o número de passos no processo com intervenção de operadores humanos é drasticamente reduzido. O CNC reduziu também o número de erros humanos (o que aumenta a qualidade dos produtos e diminui o retrabalho e o desperdício), agilizou as linhas de montagens e tornou-as mais flexíveis, pois a mesma linha de montagens pode agora ser adaptada para produzir outro produto num tempo muito mais curto do que com os processos tradicionais de produção. Acompanhando o desenvolvimento tecnológico da informática e a tendência por uma interatividade cada vez maior com o usuário, o código e linguagem de máquina também evoluíram.

2.2 Produtos Existentes no Mercado

Essa seção exibirá algumas tecnologias atuais presentes no mercado a fim de que mais tarde sirvam para análise juntamente com o projeto desenvolvido.

2.2.1 Fresadora IEC3020.

A fresadora IEC302 é fornecida pela empresa chinesa IEHK Laser & CNC Systems Co. Ltd. A Figura 3 é uma foto do produto:

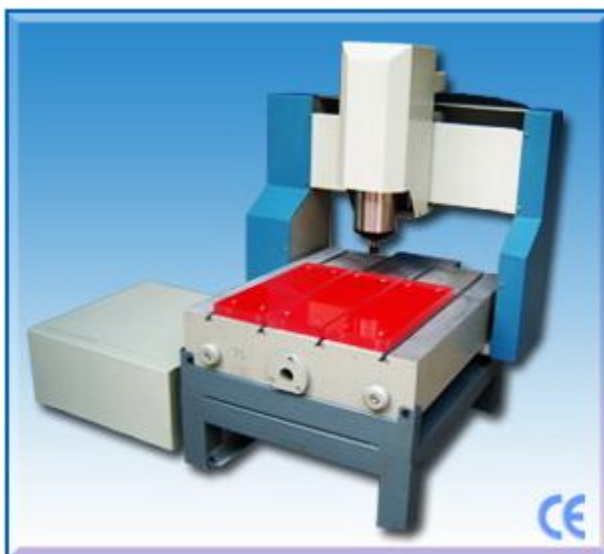


Figura 3 - Fresa IEC3020

E em seguida na Tabela 1 algumas especificações técnicas:

Especificações Técnicas	
Área Útil X, Y	300 mm x 300 mm
Interface para comunicação	Porta paralela (LTP)
Resolução	0.01mm
Velocidade de fresamento	1000 mm/min
Velocidade do motor	0-20000 rpm
Potência do motor	200w
Diâmetro da broca	4mm
Alimentação	220V
Software	Mach3
Preço	\$1200

Tabela 1 - Especificações Técnicas da Fresa IEC3020

Todas as informações foram retiradas do site da empresa IEHK LASER AND CNC Systems CO. LTD. (IEHK LASER AND CNC SYSTEMS CO. LTD., 2009).

2.2.2 Fresadora SHG0404 CNC Router

A fresadora SHG0404 CNC Router é importada pela empresa CNC Máquinas, a Figura 4 mostra a fresa:



Figura 4 - Fresa SHG0404

Na Tabela 2 abaixo algumas especificações técnicas da fresa:

Especificações Técnicas	
Área Útil X, Y, Z	400 mm x 400mm x100 mm
Interface para comunicação	USB
Resolução	< 0.01mm
Velocidade de fresamento	3000 mm/min
Velocidade do motor	6000 – 24000 rpm
Potência do motor	800w
Alimentação	220V
Preço	R\$19000

Tabela 2 – Especificações Técnicas da Fresa SHG0404

Essa fresa se destaca por também fresar em três dimensões, isto é, é possível também ter controle sobre o eixo Z, todas as informações foram coletadas no site da empresa (CNC MÁQUINAS, 2009).

2.2.3 Fresadora LPKF ProtoMat S100

A fresadora LPKF ProtoMat S100 é produzida pela empresa alemã PKF Laser & Electronics AG, a Figura 5 mostra o produto:



Figura 5 - Fresa LPKF ProtoMat S100

Na Tabela 3 abaixo algumas especificações do produto:

Especificações Técnicas	
Área Útil X, Y, Z	229 mm x 305 mm x 38 mm
Interface para comunicação	Porta USB
Resolução	0.5 μ m
Velocidade de fresamento	150 mm/min
Velocidade do motor	0-100000 rpm
Potência do motor	200w
Diâmetro da broca	1/8''
Alimentação	120/240V
Preço	\$24330

Tabela 3 - Especificações Técnicas da Fresa LPKF ProtoMat S100

As informações dessa seção foram retiradas do site da empresa LPKF Laser & Electronics (LPKF LASER & ELECTRONICS AG, 2010).

2.3 Conceitos de Linguagem de Software e Algoritmos Utilizados

2.3.1 Linguagem Visual Basic

A linguagem Visual Basic foi criada em 1991 pela empresa Microsoft e até nos dias de hoje continua proporcionando aos desenvolvedores a possibilidade de visualmente construir aplicações para desktops, telefones, navegadores e, atualmente, na nuvem (cloud computing). A versão mais atual (lançada no ano de 2010) é a Visual Basic 10, integrante da plataforma de desenvolvimento Visual Studio 2010.

Um aperfeiçoamento do BASIC, a linguagem é dirigida por eventos (event driven), e possui também um ambiente de desenvolvimento integrado (IDE - Integrated Development Environment) totalmente gráfico, facilitando enormemente a construção da interface das aplicações (GUI - Graphical User Interface), daí o nome "Visual". Em suas primeiras versões, o Visual Basic não permitia acesso a bancos de dados, sendo, portanto voltado apenas para iniciantes, mas devido ao sucesso entre as empresas - que faziam uso de componentes adicionais fabricados por terceiros para acesso a dados - a linguagem logo adotou tecnologias como DAO, RDO, e ADO, também da Microsoft, permitindo fácil acesso a bases de dados. Mais tarde foi adicionada também a possibilidade de criação de controles ActiveX, e, com a chegada do Visual Studio .NET, o Visual Basic - que era pseudo-orientada a objetos - tornou-se uma linguagem totalmente orientada a objetos (OO). (MICROSOFT CORPORATION, 2010)

No caso desse projeto, a plataforma de desenvolvimento utilizada foi o Visual Basic 6.0. Uma idéia do ambiente de desenvolvimento pode ser vista na Figura 6:

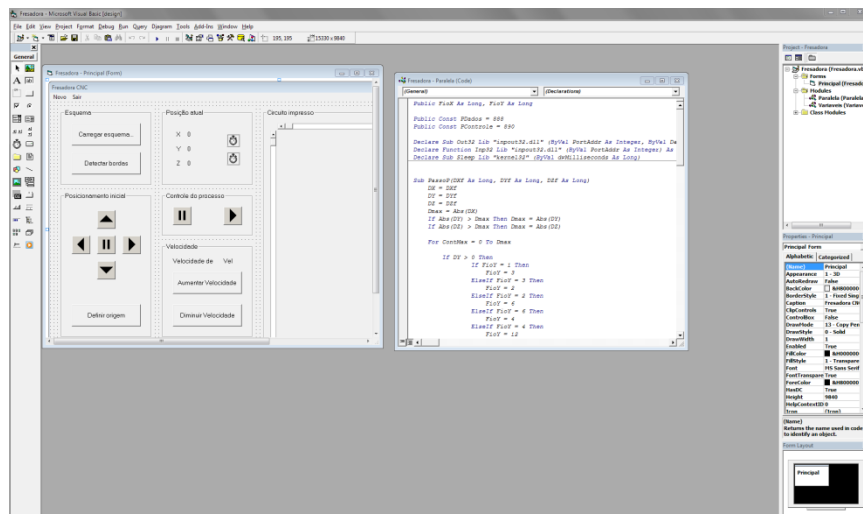


Figura 6 - Ambiente de Desenvolvimento Visual Basic 6.0

A linguagem tem estrutura similar às outras linguagens, isto é, utiliza de rotinas de repetição, decisão, desvio condicional, funções. Vale destacar o tipo de dados presente nessa linguagem. Eles são apresentados na Tabela 4:

Tipo de Dados	Descrição
Variant	Tipo genérico
Boolean	True ou False
Byte	0 até 255
Integer	-32768 até 32767
Long	-2147483648 até 2147483648
Single	-3,402823E38 até -1,401298E-45 (para valores negativos) 1,401298E-45 até 3,402823E38 (para valores positivos)
Double	-1,79769313486232E308 até -4,94065645841247E-324 (negativos) 4,94065645841247E-324 até 1,79769313486232E308 (positivos)
Currency	-922.337.203.685.477,5808 até 922.337.203.685.477,5807
Decimal	+/-79.228.162.514.264.337.593.543.950.335 (sem casas decimais) +/-7,9228162514264337593543950335 (com casas decimais)
Date	1 de Janeiro de 100 até 31 de Dezembro de 9999
String	1 até aproximadamente 2 bilhões de caracteres, 65.400 caracteres se tamanho fixo
Type	Definido pelo utilizador
Object	Referência a objetos

Tabela 4 - Tipos de Dados da Linguagem Visual Basic

Para o entendimento da linguagem e dos arquivos gerados ao se criar o projeto (exemplo: Figura 7 que contém os arquivos do projeto), na Figura 7 estão listados os tipos de arquivos que o Visual Basic gera ao se compilar um projeto.




























 BotãoBaixoAtivo.bmp	26/11/2009 1:30 PM	Bitmap image	5 KB
 BotãoBaixoInativo.bmp	26/11/2009 1:57 PM	Bitmap image	5 KB
 BotãoCimaAtivo.bmp	26/11/2009 1:30 PM	Bitmap image	5 KB
 BotãoCimaInativo.bmp	26/11/2009 1:57 PM	Bitmap image	5 KB
 BotãoContinuarAtivo.bmp	26/11/2009 1:30 PM	Bitmap image	5 KB
 BotãoContinuarInativo.bmp	26/11/2009 1:56 PM	Bitmap image	5 KB
 BotãoDireitaAtivo.bmp	26/11/2009 1:30 PM	Bitmap image	5 KB
 BotãoDireitaInativo.bmp	26/11/2009 1:56 PM	Bitmap image	5 KB
 BotãoEsquerdaAtivo.bmp	26/11/2009 1:30 PM	Bitmap image	5 KB
 BotãoEsquerdaInativo.bmp	26/11/2009 1:57 PM	Bitmap image	5 KB
 BotãoPararAtivo.bmp	29/11/2009 10:31 AM	Bitmap image	5 KB
 BotãoPararInativo.bmp	29/11/2009 10:36 AM	Bitmap image	5 KB
 BotãoPassoAtivo.bmp	26/11/2009 1:43 PM	Bitmap image	5 KB
 BotãoPassoInativo.bmp	26/11/2009 1:56 PM	Bitmap image	5 KB
 BotãoPausaAtivo.bmp	26/11/2009 1:40 PM	Bitmap image	5 KB
 BotãoPausaInativo.bmp	26/11/2009 1:53 PM	Bitmap image	5 KB
 Fresadora.frm	7/12/2009 12:09 PM	Visual Basic Form ...	30 KB
 Fresadora.frx	7/12/2009 12:09 PM	Visual Basic Form ...	100 KB
 Fresadora.vbp	7/12/2009 12:09 PM	Visual Basic Project	2 KB
 Fresadora.vbw	21/9/2010 9:05 AM	Visual Basic Projec...	1 KB
 GCommonDialog.cls	18/11/2009 7:27 PM	Visual Basic Class ...	42 KB
 Ícone.ico	19/11/2009 1:00 AM	Icon	3 KB
 inpout32.dll	22/7/2003 11:04 PM	Application extens...	32 KB
 MSSCCPRJ.SCC	19/11/2009 12:29 PM	Microsoft SourceS...	1 KB
 Paralela.bas	5/12/2009 3:23 AM	Visual Basic Module	4 KB
 Variaveis.bas	7/12/2009 12:05 PM	Visual Basic Module	3 KB
 VB76.tmp	22/11/2009 11:10 AM	TMP File	3 KB

Figura 7 - Exemplo de Arquivos Gerados pelo Visual Basic 6

Tipo de Arquivo	Descrição
.vdp	De Projeto
.frm	De Formulário
.bas	De Módulos
.ocx	De Controles ActiveX
.ctl	De Controles do Usuário
.pag	De páginas de propriedades
.res	De recursos
.exe, .dll e outros	Externos

Tabela 5- Tipos de Arquivos Utilizados pelo Visual Basic 6

2.3.2 Conceitos de Visão Computacional

O principal conceito envolvido no projeto, relacionado à Visão Computacional seria o Thresholding. Trata-se de um valor que serve como limite, no caso do formato de imagem utilizado no algoritmo (BITMAP de 24 bits) as cores são divididas em RGB (Red Green Blue), cada cor tem

uma graduação que vai de 0 (Zero) a 255. A imagem que vemos na tela nada mais é que um composto da mistura das 3 cores. Thresholding pode ser melhor entendido se utilizarmos a palavra “Binarização”, pois trata-se de um valor definido (manualmente ou automaticamente) que serve como divisor, isto é, acima desse valor é definido como 255 (valor máximo, cor branca) ou 0 (valor mínimo, preto).

Além desse conceito é importante citar outros métodos de filtragem de bordas que serão utilizados como ferramentas para comparação através do software Matlab da empresa MathWorks. Serão esses os métodos utilizados:

- Thresholding automático;
- Detector de Sobel;
- Detector de Roberts;
- Detector de Prewitt;
- Detector de Canny;

Todos esses métodos, bem como as sintaxes em Matlab podem ser vistos na seção de Anexos correspondente.

2.4 Considerações Finais

Esse capítulo apresentou alguns conceitos chave para o entendimento do restante da monografia. Destacam-se os tópicos de tecnologias existentes pois esses dados serão fundamentais para as análises posteriores dos resultados obtidos.

O processo de desenvolvimento para o alcance dos resultados pode ser visto no capítulo que se segue.

Capítulo 3 – Materiais e Métodos

Esse capítulo descreve o processo de desenvolvimento do protótipo e software para que assim seja possível adquirir os dados necessários para análise através das metodologias adotadas descritas também nesse capítulo.

3.1 Materiais Utilizados

3.1.1 Peça de Hardware

Essa seção abordará os aspectos construtivos da estrutura de hardware empregada, mais precisamente a estrutura que foi fabricada através de processo metalúrgico.

Para o início do projeto algumas premissas devem ser tomadas, uma delas foi o a área de atuação da fresa. Nesse caso optou-se por uma área de no máximo o tamanho de uma folha de sulfite A4, isto é, 120 mm por 355,6 mm. Esse parâmetro é um dos principais para definição das dimensões do protótipo.

Na Figura 8 pode-se ver uma foto do protótipo pronto.



Figura 8 - Protótipo da Fresa desenvolvida

A seguir estão listados os principais componentes do hardware, acessórios como parafusos, porcas, arruelas não foram listados. Para facilitar o entendimento, o hardware será dividido em módulos e adota-se orientação conforme a Figura 8.

- Base:

Parte	Quantidade (un.)	Parâmetros
Barras horizontais	2	46 x 1,5 x 1,5 cm
Barras verticais	2	26 x 1,5 x 1,5 cm
Trilhos Lisos	2	29 cm (1cm de diâmetro)
Trilho Rosqueados	1	30 cm (1cm de diâmetro)
Pés	4	4 x 1,5 x 1,5 cm
Engrenagem	1	4 cm de diâmetro
Suporte para o motor	1	4 x 1,5 x 1,5 cm
Motor de Passo 5 fios	1	12 V

Tabela 6 - Descrição da Base do Hardware

- Plataforma

Parte	Quantidade (un.)	Parâmetros
Bandeja	1	42 x 52 cm
Suporte para o trilho liso	4	2 cm de diâmetro
Suporte para o trilho Rosqueado	1	2 cm de diâmetro

Tabela 7 - Descrição da Plataforma do Hardware

- Suporte do cabeçote fresador

Parte	Quantidade (un.)	Parâmetros
Barras horizontais	1	46 x 1,5 x 1,5 cm
Barras verticais	2	13 x 1,5 x 1,5 cm
Trilhos Lisos	2	46 cm (1cm de diâmetro)
Trilho Rosqueados	1	30 cm (1cm de diâmetro)
Engrenagem	1	4 cm de diâmetro
Suporte para o motor	1	4 x 1,5 x 1,5 cm
Motor de Passo 5 fios	1	12 V

Tabela 8 - Descrição do Suporte do Cabeçote Fresador do Hardware

O cabeçote fresador ainda não foi desenvolvido. No lugar desenvolveu-se um suporte para uma caneta, como esse não é o objetivo, não haverá descrição para essa parte do hardware.

O hardware completo, sem considerar a mão de obra, custou por volta de 350 reais.

3.1.2 Aplicativo Desenvolvido

Como citado na seção 2.3.1 foi utilizado a plataforma Visual Basic 6.0 para o desenvolvimento do software.

Um dos objetivos do projeto é a simplicidade e acessibilidade. Por isso foi escolhido a linguagem Visual Basic. Os requisitos para o funcionamento são basicamente os mesmos requisitos que um computador precisa pra rodar o Windows XP:

Componente	Mínimo	Recomendado
Processador	233 MHz	300 MHz ou maior
Memória	64 MB RAM	128 MB RAM ou maior
Adaptador de Vídeo e Monitor	Super VGA (800 x 600)	Super VGA (800 x 600) ou superior
Controles	Teclado e Mouse	Teclado e mouse

Tabela 9 - Requisitos para o Software da Fresa

Abaixo na Figura 9 tem-se a interface final do software desenvolvido.

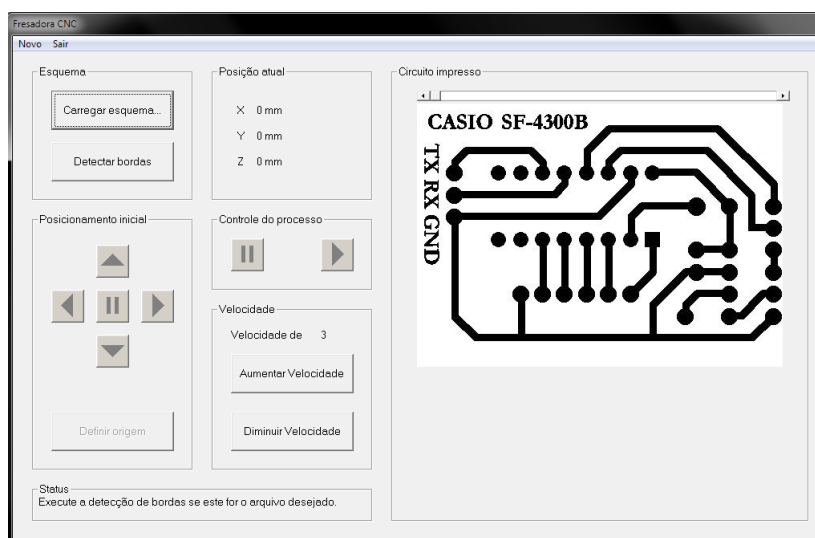


Figura 9 - Interface do Programa da Fresa

As etapas para utilização do software são sequenciais, para esclarecer as etapas o Digrama de Fluxo de Dados do Projeto (Figura 10) está disposto com uma breve síntese dos principais procedimentos e suas relações.

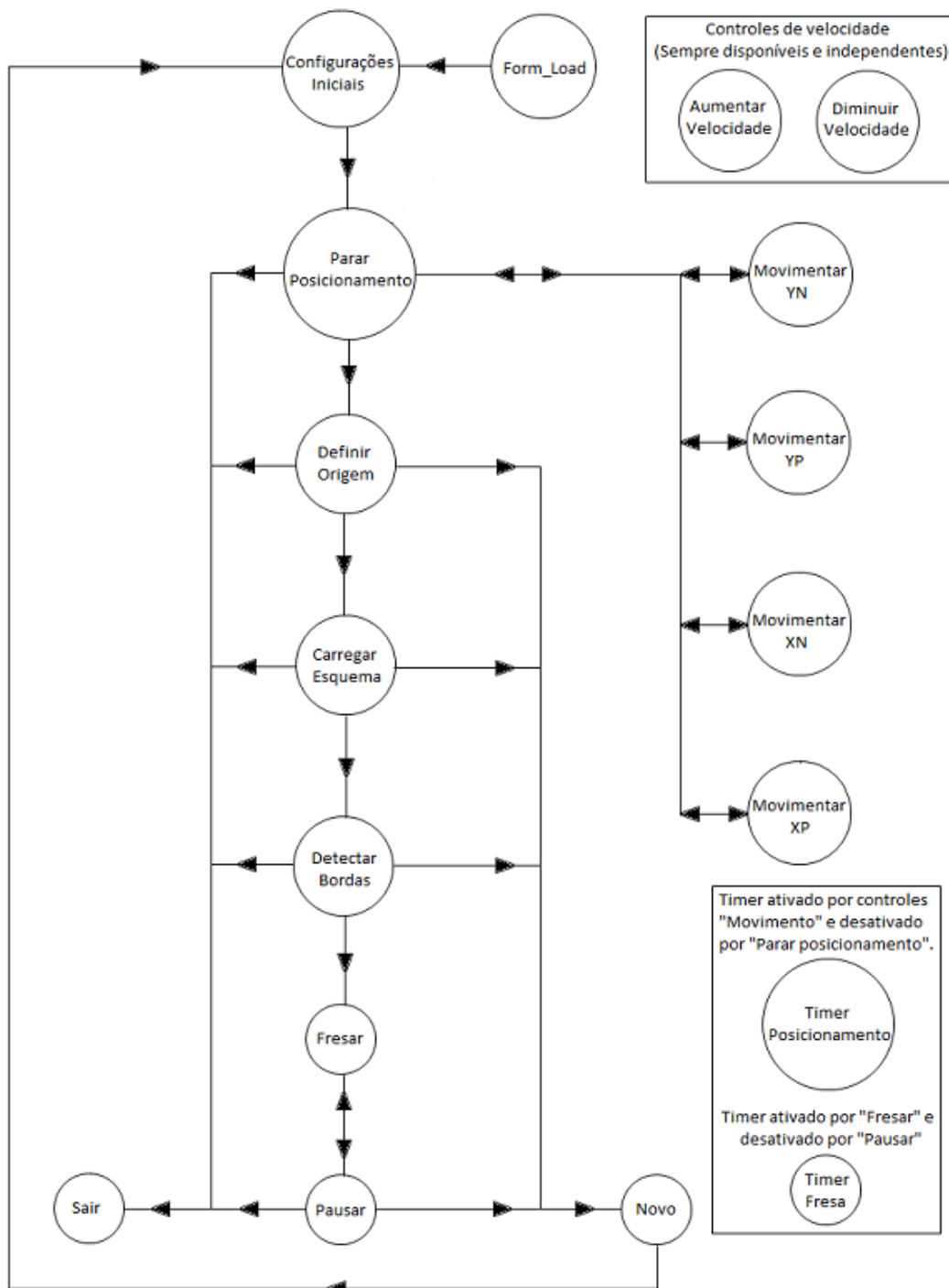


Figura 10 - Diagrama de Fluxo de Dados do Programa

Os códigos referentes ao programa podem ser vistos na seção de Apêndices (Apêndice A, Apêndice B e Apêndice C). Está no apêndice também, um pseudo-código para facilitar o entendimento do algoritmo, nele é extraído a idéia principal de cada linha do código e explicada em uma linguagem mais simples compreensível por quem não conheça a linguagem VB.

Basicamente o processo visto pela interface pode ser resumido da seguinte forma (considerando um procedimento de sucesso, as outras formas de lidar com o programa estão descrito na Figura 10): Primeiramente o usuário define a origem da placa através dos cursores de movimento da fresa (Figura 11). A origem da placa é o ponto inicial da fresagem e deve ser apontado pelo usuário como o canto superior direito da placa para que a fresagem ocorra corretamente.

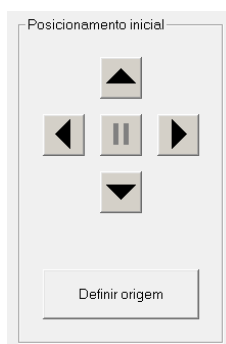


Figura 11 - Cursor de Movimentação da Fresa

Após definido a origem, o usuário carrega uma imagem com o formato BITMAP de 24bits. Caso a imagem não esteja com esse padrão o programa reconhecerá e apontará um erro na janela de STATUS. Carregada a imagem, dá-se o comando “Detectar Bordas” e o resultado é mostrado na Figura 15.

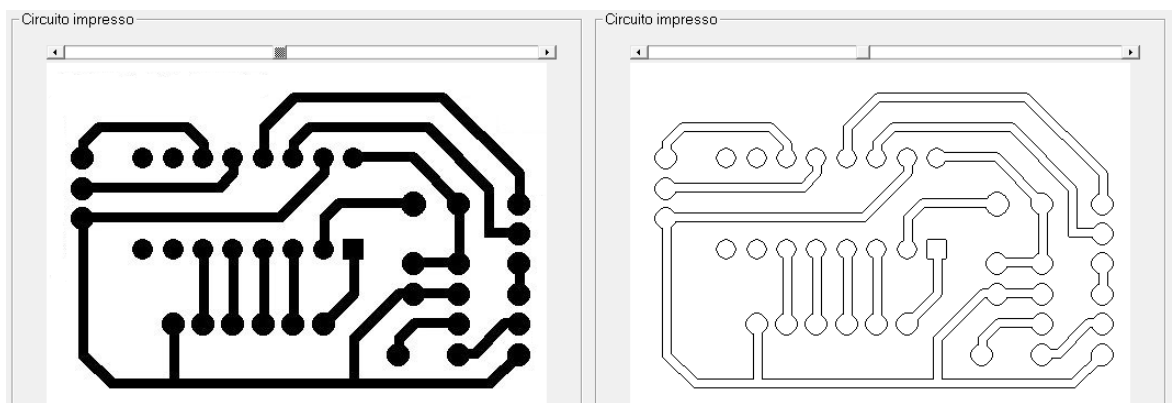


Figura 12 - Antes e Depois da Detecção das Bordas

A próxima etapa é definir a velocidade da fresa (quanto maior a velocidade menor a garantia de precisão e qualidade) e iniciar a fresagem através do painel de controle do processo (Figura 13).



Figura 13 - Painel de Controle do Processo

Note que na caixa onde o esquema é carregado a imagem muda para a cor vermelha conforme a fresagem está sendo feita (tempo-real). A posição da fresa é mostrada no quadro de posição (Figura 14).

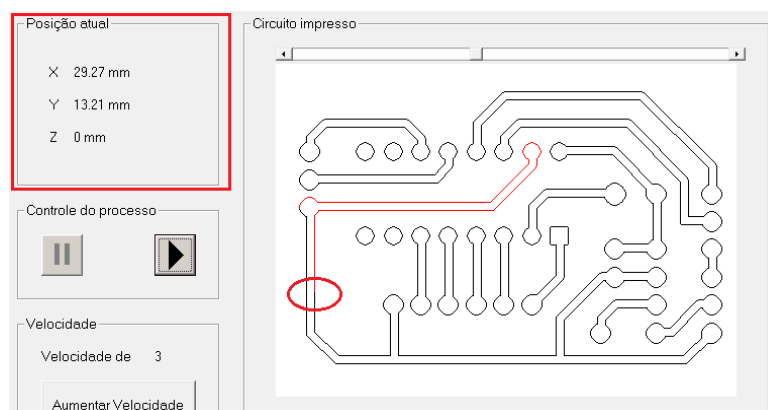


Figura 14 - Processo de Fresagem em Tempo Real

Assim é possível sintetizar o processo como um todo.

3.1.3 Codificação

Dentro do código, como é possível ver na Figura 15, temos um exemplo de uma parte do programa, cada divisão corresponde a um processo. Cada um deles será descrito em seguida justamente com as variáveis envolvidas em cada bloco.

```
.....  
.....  
'  
..... Funções dos Botões de Controle do Processo  
.....  
.....  
  
Public Sub hscScroll_Scroll()  
    hscScroll_Change  
End Sub  
  
Public Sub hscScroll_Change()  
    EsquemaOriginal.Left = -Screen.TwipsPerPixelY * hscScroll.Value  
End Sub  
  
Public Sub vscScroll_Change()  
    EsquemaOriginal.Top = -Screen.TwipsPerPixelY * vscScroll.Value  
End Sub  
  
Public Sub vscScroll_Scroll()  
    vscScroll_Change  
End Sub  
  
Public Sub picScrollBar_Resize()  
    On Error Resume Next  
    hscScroll.Visible = (picScrollBar.ScaleWidth - vscScroll.Width < EsquemaOriginal.Width)  
    vscScroll.Visible = (picScrollBar.ScaleHeight - hscScroll.Height < EsquemaOriginal.Height)  
    If (hscScroll.Visible) Then  
        hscScroll.Max = (EsquemaOriginal.Width - picScrollBar.ScaleWidth + vscScroll.Width * Abs(vscScroll.Visible)) \ Screen.TwipsPerPixelX  
        hscScroll.SmallChange = 32  
    End If  
    If (vscScroll.Visible) Then  
        vscScroll.Max = (EsquemaOriginal.Height - picScrollBar.ScaleHeight + hscScroll.Height * Abs(hscScroll.Visible)) \ Screen.TwipsPerPixelY  
        vscScroll.SmallChange = 32  
    End If  
End Sub
```

Figura 15 - Trecho do Código na Tela do Compilador

- Procedimento Form_Load

Este procedimento define valores iniciais de algumas variáveis globais e escreve valores em alguns mostradores da tela.

Variáveis globais modificadas: FioX, FioY, XP, YP, ZP, NumeroRepeticoes

- Procedimento ConfiguracoesIniciais

Este procedimento define ativa ou desativa objetos do programa e confirma se os motores estão desligados.

Variáveis globais modificadas: InterrupcaoRastreador

- Procedimento BotaoDefOrigem_Click

Este procedimento define os valores das variáveis globais no novo ponto de referência, assim como ativa e desativa objetos.

Variáveis globais modificadas: XC, YC, ZC, XCA, YCA, ZCA, XI, YI, ZI, XP, YP, ZP

- Procedimento BotaoMovimentarXN_Click
Este procedimento ajusta ativa o movimento do cabeçote na direção $-x$.
Variáveis globais modificadas: PassoX

- Procedimento BotaoMovimentarXP_Click
Este procedimento ajusta ativa o movimento do cabeçote na direção $+x$.
Variáveis globais modificadas: PassoX

- Procedimento BotaoMovimentarYN_Click
Este procedimento ajusta ativa o movimento do cabeçote na direção $-y$.
Variáveis globais modificadas: PassoY

- Procedimento BotaoMovimentarYP_Click
Este procedimento ajusta ativa o movimento do cabeçote na direção $+y$.
Variáveis globais modificadas: PassoY

- Procedimento BotaoPararEsp_Click
Este procedimento pára os movimentos iniciados pelos procedimentos acima.
Variáveis globais modificadas: PassoX, PassoY, PassoZ

- Procedimento Fresar_Click
Este procedimento ativa o timer e inicia o processo de fresagem.

- Procedimento Pausar_Click
Este procedimento pausa o processo de fresagem.

- Procedimento BotaoProcessar_Click
Este procedimento detecta as bordas da imagem carregada.
Variáveis globais modificadas: Auxiliar1,Auxiliar2,Original,x,y

- Procedimento BotaoCarregar_Click
Este procedimento carrega o arquivo desejado e o vincula à uma matriz.
Variáveis globais modificadas: sFile, bContinue, bmp1, sa1, Original

- Procedimento MenuNovo_Click
Este procedimento retorna o programa às condições iniciais para um novo processo.

- Procedimento MenuSair_Click
Este procedimento sair do programa.

- Procedimento DiminuirVelocidade_Click
Este procedimento diminui a velocidade dos motores de passo aumentando o número de repetições de cada comando.
Variáveis globais modificadas: NumeroRepeticoes

- Procedimento AumentarVelocidade_Click
Este procedimento aumenta a velocidade dos motores de passo diminuindo o número de repetições de cada comando.
Variáveis globais modificadas: NumeroRepeticoes

- Procedimento hscScroll_Change
Este procedimento trata da barra de rolagem horizontal.

- Procedimento vscScroll_Change
Este procedimento trata da barra de rolagem vertical.

- Procedimento hscScroll_Click
Este procedimento trata da barra de rolagem horizontal.

- Procedimento vscScroll_Click
Este procedimento trata da barra de rolagem vertical.

- Procedimento picScrollBar_Resize
Este procedimento ajusta as escalas das barras de rolagem.

- Procedimento TimerPosicionamento_Timer
Este procedimento é executado periodicamente durante o posicionamento inicial da fresa.

- Procedimento TimerFresa_Timer

Este procedimento é executado periodicamente durante a fresagem.

- Procedimento Final

Este procedimento ajusta os objetos do programa após o término da fresagem.

- Procedimento ProcessarCNC

Este procedimento encontra traços contínuos na imagem e movimenta ao longo deles.

Variáveis globais modificadas: x, y, XI, YI, InterrupcaoRastreador, Auxiliar2, XC, YC

- Procedimento Movimentar

Este procedimento calcula o próximo deslocamento da fresa, envia o pedido ao procedimento adequado, e identifica em cor vermelha os trechos já percorridos.

Variáveis globais modificadas: XCA, YCA, ZCA, XC, YC, ZC, Original

- Procedimento Rastreador

Este procedimento segue uma borda contínua na figura e determina as variações de pixels para depois convertê-los em passos.

Variáveis globais modificadas: PixelsDesenhados, InterrupcaoRastreador

Uma importante etapa a ser destacada é a parte de detecção de bordas do esquema, a próxima seção é dedicada a esse procedimento.

3.1.4 Algoritmo de Detecção das Bordas

Uma etapa fundamental no processo é a detecção de bordas, pois é a partir do resultado desse procedimento que as instruções para os motores serão geradas. Pode-se ver na Figura 12 o resultado do procedimento.

Como visto na seção 2.3.2 foi utilizado o princípio de Threshold. O algoritmo utilizado pode ser visto no Apêndice C na parte “Função de Detecção de Bordas”.

Basicamente o algoritmo transforma a imagem carregada pelo usuário em uma matriz e armazena cada pixel em uma posição. Mais tarde essa matriz é totalmente percorrida e o algoritmo analisa se o pixel possui um valor maior ou menor que 128. Se maior ele joga o pixel para branco (valor 255) ou se menor joga para preto (valor 1). Ao mesmo tempo em que essa verificação ocorre uma lógica de armazenamento da borda também acontece, fazendo com que o resultado seja apenas

a borda da imagem (Figura 16) com a largura de 1 pixel, assim a fresa necessita passar somente uma vez a cada trecho.

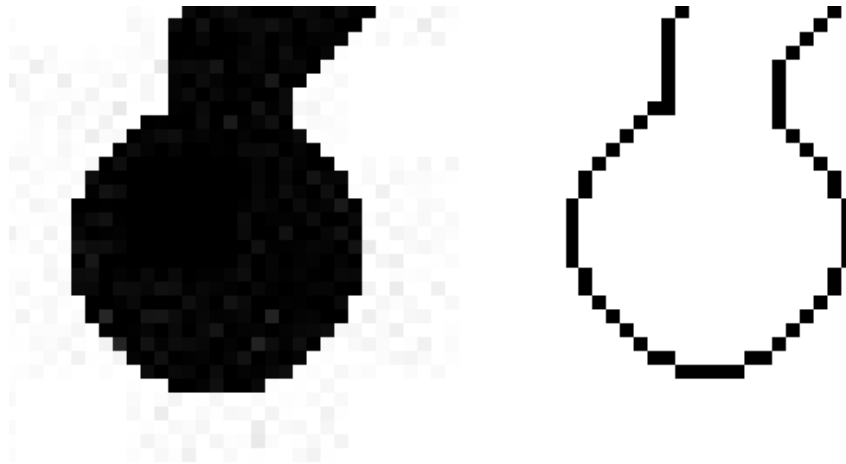


Figura 16 - Resultado do Algoritmo de Detecção de Bordas

Mais tarde o algoritmo desenvolvido será comparado com algoritmos já conhecidos na literatura e será aplicado também em imagens coloridas para verificar a eficiência do código.

3.1.5 Circuito de Potência da Impressora

A comunicação do software com hardware é feito por meio da porta paralela do computador (LTP).

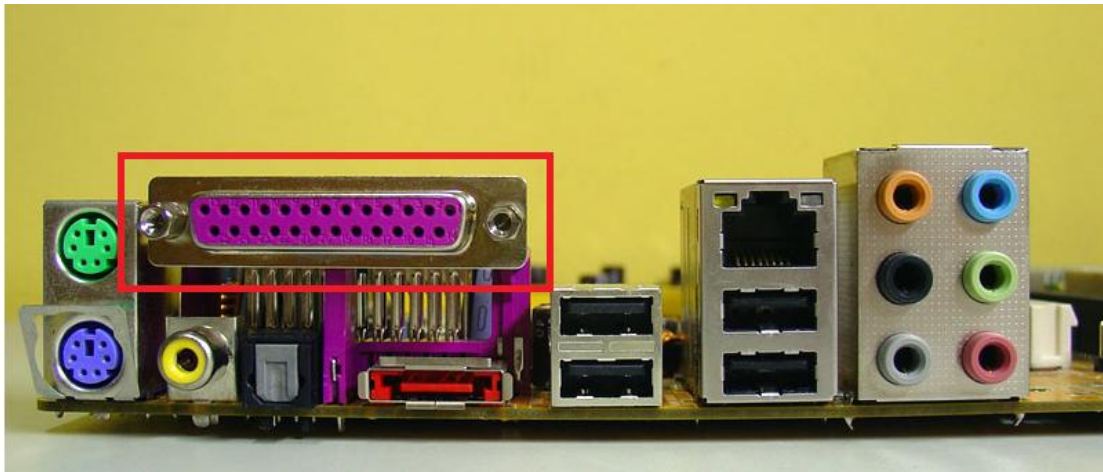


Figura 17- Porta Paralela

Como podemos ver na Figura 17 o conector padrão da porta é o DB25, a comunicação pode ser feita de duas formas: Unidirecionalmente e bidirecionalmente.

No caso do projeto a escolha foi pela forma unidirecional, com essa definição a porta paralela SPP (Standard Parallel Port) 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 (MESSIAS, 2006).

O funcionamento no modo SPP se dá da seguinte forma (Figura 18):

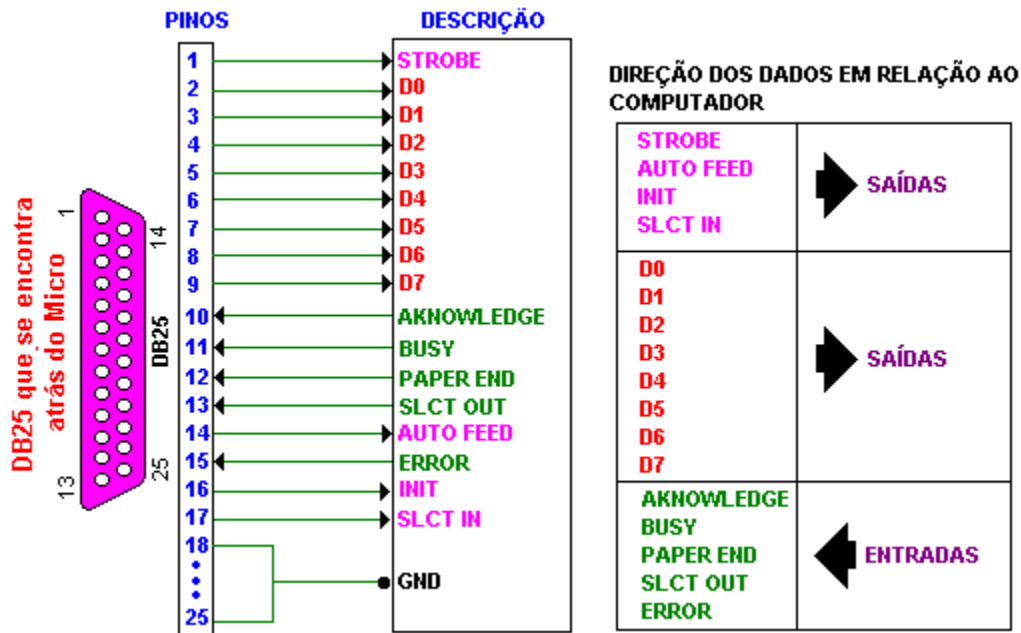


Figura 18 - Conector DB25 da Porta Paralela no modo SPP

Portanto, pode-se escolher qualquer uma das saídas de dados para servir para comunicação entre hardware e software. Um problema que aparece é a baixa corrente disponibilizada pela porta paralela (2,6mA), por isso é necessário um circuito que forneça as correntes necessárias para fazer os motores girarem. O esquema proposto é simples e está representado por um circuito equivalente na Figura 19:

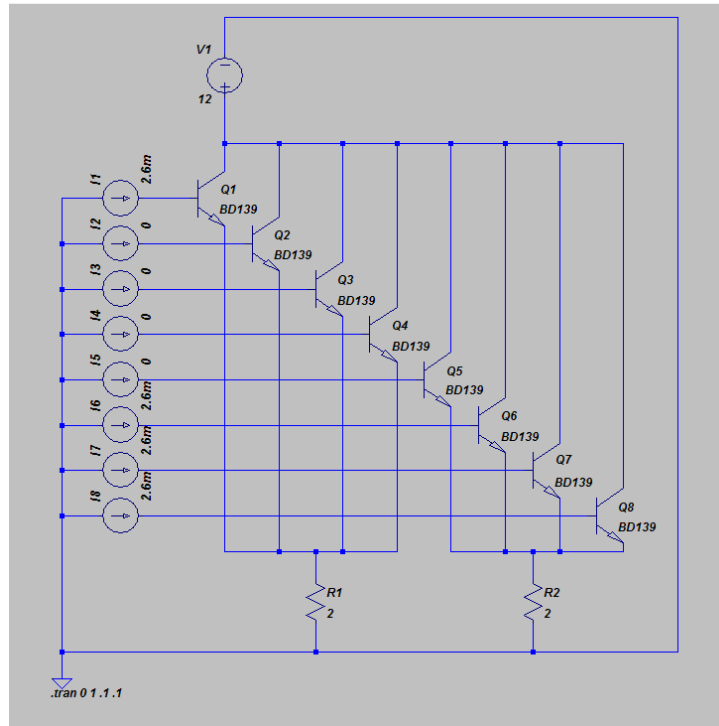


Figura 19 - Equivalente eletrônico da Fresa

Na Figura 19 está representado a porta paralela pelas fontes de corrente e o terra na parte esquerda da figura e os motores de passo como resistências de 20hms. Cada motor de passo possui 4 fios que alimentam as bobinas que geram o campo e um fio comum. A fonte utilizada foi uma fonte de 12V e 3A.

3.2 Metodologia

A fim de justificar os objetivos através dos material descrito na seção 3.1 serão adotados métodos descritos em seguida.

3.2.1 Testes de Limites do Sistema Projetado

Os limites do sistema projetado são fundamentais para compararmos o sistema com o que há atualmente no mercado. Para isso é necessário checar parâmetros como:

- Velocidade de deslocamento da fresa (velocidade de fresamento)

Para se medir a velocidade de fresamento o método será imprimir em uma folha de sulfite um quadrado de tamanho 10x10cm com o seguinte formato (Figura 20):

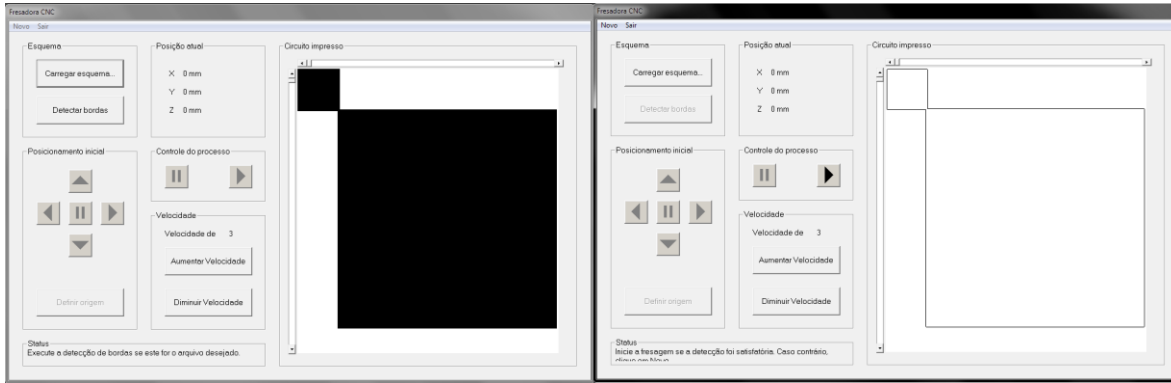


Figura 20 - Imagem utilizada para Determinação de Velocidade de Deslocamento

A figura adjacente serve para determinar o momento em que se deve disparar e parar o cronômetro. A impressão começa no canto superior esquerdo, portanto no momento onde a fresa muda sua direção de deslocamento para começar o quadrado maior é quando se deve disparar o cronômetro. Esse tempo é o único parâmetro necessário para determinar a velocidade através da Equação 1:

$$V_f = \frac{400}{t} \left[\frac{mm}{min} \right]$$

Equação 1 - Equação da Velocidade de Fresamento

Serão feitas 15 repetições em cada uma das velocidades disponíveis (5 ao todo). As repetições permitem obter um resultado mais satisfatório.

- Área útil de impressão

A área útil de impressão pode ser definida utilizando os controles de posicionamento. Quando chegam a seus limites, os motores não avançam mais, assim, a medida dos lados da figura obtida através desse processo é a área útil de impressão.

- Resolução de impressão (Precisão)

A precisão exigida de uma Máquina Ferramenta com Comando Numérico Computadorizado – CNC como no caso da Fresadora CNC, depende da aplicação exigida (MANFÉ, 1989).

A resolução foi determinada fazendo o programa rodar com uma imagem que possui 1 pixel, isto é, o menor movimento realizado pela fresa.

3.2.2 Comparativo do Algoritmo de Detecção de Bordas

Será utilizado o Software Matlab da empresa MathWorks para que o algoritmo de detecção de bordas seja comparado com os algoritmos citados na seção 2.3.2.

Além de comparativos utilizando imagens padrões de circuitos, serão feitos testes para imagens coloridas.

3.2.3 Consumo de Energia

Para medição do consumo de energia será utilizado um Amperímetro ligado direto na saída da fonte de alimentação, assim através da Equação 2 podemos adquirir a potência consumida pela fresa.

$$P = U \cdot I \text{ [W]}$$

Equação 2 - Cálculo da Potência

Sabendo que:

$$U = 12v$$

Conseguimos obter o valor da potência da fresa.

Como temos 2 motores no total, a medição será feita em 3 momentos distintos utilizando a Figura 21:

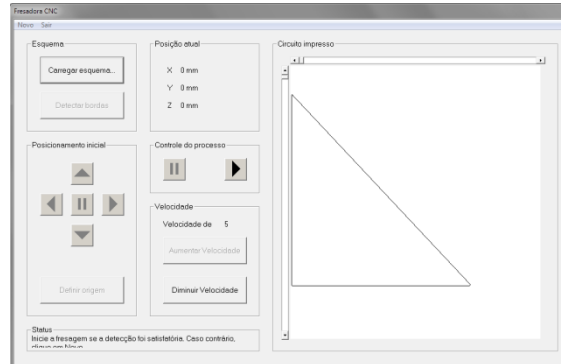


Figura 21 - Triângulo para Medição da Potência

Serão aferidas as correntes em cada lado do triângulo, no cateto inferior o motor 1 está operando e o 2 está parado, no cateto lateral o inverso e na hipotenusa ambos funcionam.

O programa dispõe de 5 velocidades de fresamento, essa variável será considerada nessa etapa.

Capítulo 4 – Resultados e Conclusões

Este capítulo irá mostrar todos os resultados obtidos dos métodos citados no capítulo 3 e as conclusões tiradas a respeito. Para todos os testes foi utilizado uma caneta (Figura 22) no cabeçote fresador de forma que a fresa imprimisse o desenho no papel em vez de marcar a placa de cobre. Podemos ver o processo na Figura 23.



Figura 22- Caneta Utilizada para os Testes de Resolução



Figura 23 - Foto do procedimento final utilizando a caneta

4.1 Testes dos limites do sistema projetado

4.1.1 Velocidade de Deslocamento da fresa

A velocidade foi determinada utilizando o método descrito no item 3.2.1.

As medições adquiridas no processo estão na Tabela 10, Tabela 11 e Tabela 12.

Velocidade 1				Velocidade 2			
Iteração	Tempo em Segundos	Tempo em Minutos	Vf	Iteração	Tempo em Segundos	Tempo em Minutos	Vf
1	72.3	1.205	331.9502	1	59.5	0.991666667	403.3613
2	72.1	1.201666667	332.871	2	60	1	400
3	72	1.2	333.3333	3	59.3	0.988333333	404.7218
4	72.4	1.206666667	331.4917	4	58.5	0.975	410.2564
5	73	1.216666667	328.7671	5	60.5	1.008333333	396.6942
6	72.1	1.201666667	332.871	6	60	1	400
7	73	1.216666667	328.7671	7	60.4	1.006666667	397.351
8	73.1	1.218333333	328.3174	8	59.5	0.991666667	403.3613
9	72.1	1.201666667	332.871	9	59.8	0.996666667	401.3378
10	72.2	1.203333333	332.41	10	60	1	400
11	72.1	1.201666667	332.871	11	60.3	1.005	398.01
12	72	1.2	333.3333	12	59.7	0.995	402.0101
13	73.2	1.22	327.8689	13	58	0.966666667	413.7931
14	72.3	1.205	331.9502	14	60	1	400
15	72	1.2	333.3333	15	59.2	0.986666667	405.4054
Vf Média			331.5338	Vf Média			402.4202

Tabela 10 - Velocidade de Deslocamento da Fresa (1)

Velocidade 3				Velocidade 4			
Iteração	Tempo em Segundos	Tempo em Minutos	Vf	Iteração	Tempo em Segundos	Tempo em Minutos	Vf
1	45.6	0.76	526.3158	1	30	0.5	800
2	44.6	0.7433333	538.1166	2	31.2	0.52	769.2308
3	46	0.7666667	521.7391	3	30.5	0.5083333	786.8852
4	44	0.7333333	545.4545	4	31	0.5166667	774.1935
5	44.8	0.7466667	535.7143	5	30	0.5	800
6	45.2	0.7533333	530.9735	6	30.4	0.5066667	789.4737
7	44.5	0.7416667	539.3258	7	31.2	0.52	769.2308
8	45	0.75	533.3333	8	30	0.5	800
9	44.3	0.7383333	541.7607	9	30.5	0.5083333	786.8852
10	43.8	0.73	547.9452	10	31	0.5166667	774.1935
11	44	0.7333333	545.4545	11	31.2	0.52	769.2308
12	44	0.7333333	545.4545	12	30.8	0.5133333	779.2208
13	44.1	0.735	544.2177	13	29.8	0.4966667	805.3691
14	45	0.75	533.3333	14	30	0.5	800
15	45.3	0.755	529.8013	15	31	0.5166667	774.1935
Vf Média			537.2627	Vf Média			785.2071

Tabela 11 - Velocidade de Deslocamento da Fresa (2)

Velocidade 5			
Iteração	Tempo em Segundos	Tempo em Minutos	Vf
1	16.3	0.271666667	1472.393
2	15.5	0.258333333	1548.387
3	16	0.266666667	1500
4	16.7	0.278333333	1437.126
5	15.7	0.261666667	1528.662
6	16.2	0.27	1481.481
7	16	0.266666667	1500
8	16.8	0.28	1428.571
9	15.6	0.26	1538.462
10	16.3	0.271666667	1472.393
11	16	0.266666667	1500
12	17	0.283333333	1411.765
13	16.7	0.278333333	1437.126
14	16.5	0.275	1454.545
15	16	0.266666667	1500
		Vf Média	1480.727

Tabela 12 - Velocidade de Deslocamento da Fresa (3)

4.1.2 Área útil de impressão

Foi realizado um procedimento onde o cabeçote fresador partiu de um vértice até o vértice oposto, assim os valores dos segmentos foram:

	Limites
Horizontal	350,2 mm
Vertical	124,7 mm

Tabela 13 - Área útil de impressão

4.1.3 Resolução de impressão

O resultado da impressão de 1 pixel foi um movimento quase imperceptível, já que o movimento de um passo já é pequeno e esse movimento ainda é transmitido pela barra rosqueada do trilho. Pode-se dizer, portanto, que a resolução é menor que 1mm.

4.1.4 Conclusões e comparativo com outras tecnologias

A Tabela 14 será composta com os dados adquiridos nas seções passadas e dos dados coletados nas tabelas da seção 2.2.

Parâmetro	IEC3020	SHG0404	LPKF ProtoMat S100	Fresa Projetada
Velocidade da Fresa	1000 mm/min	3000 mm/min	150 mm/min	400 a 1480 mm/min
Área Útil	300 mm x 300 mm	400 mm x 400mm x100 mm	229 mm x 305 mm x 38 mm	350mm x 125mm
Resolução	0.01mm	< 0.01mm	0.5 µm	<1mm

Tabela 14 - Comparativo das Tecnologias

Analisando os dados percebe-se que o sistema projetado, em comparação com os outros, possui parâmetros que não fogem muito no nível de qualidade das outras tecnologias, salientando que o projeto é destinado ao público amador, percebe-se que possui uma boa velocidade de fresamento, uma área útil similar e uma resolução adequada para circuitos discretos. A resolução não pode ser medida com precisão pois não foi possível encontrar um método adequado para tal.

4.2 Comparativo do algoritmo de detecção de bordas

4.2.1 Resultados

Nas figuras abaixo pode-se ver, lado a lado, os resultados de cada método aplicado para diversos tipos de imagens.

Primeiramente foram aplicados na Figura 24 os filtros de Sobel, Prewitt, Roberts, Canny e o algoritmo desenvolvido. Após o algoritmo aplicou-se também um nível de tresholding para que houvesse a binarização da imagem obtida pelos filtros.







 <p>Figura 24 - Imagem Original</p>	 <p>Figura 25 - Sobel e Threshold</p>	 <p>Figura 26 - Prewitt e Threshold</p>
 <p>Figura 27 - Roberts e Threshold</p>	 <p>Figura 28 - Canny e Threshold</p>	 <p>Figura 29 - Algoritmo desenvolvido para fresa</p>

Tabela 15 - Comparação dos Filtros de Detecção de Bordas para uma Imagem Qualquer

Sabendo que a principal funcionalidade é a detecção de bordas de esquemas eletrônicos, aplicou-se o filtro de Roberts (uma vez que para esse tipo de imagem o resultado dos demais filtros são semelhantes ao obtido) e um nível de threshold para binarização. Os resultados para o esquema da Figura 12 são apresentados na Figura 30 e Figura 31.

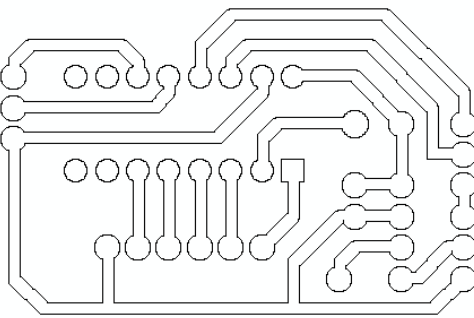
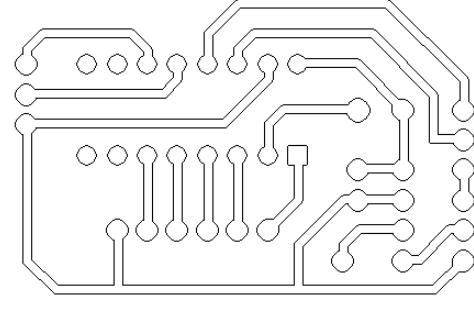
 <p>Figura 30 - Resultado do Filtro de Roberts e o Threshold</p>	 <p>Figura 31 - Resultado do algoritmo desenvolvido</p>
--	--

Tabela 16 - Comparação do Filtro de Roberts com o algoritmo desenvolvido para um esquema de circuito

Estendendo um pouco a aplicação do algoritmo, realizou-se uma análise do seu comportamento para figuras coloridas, uma vez que o programa aceita esse tipo de entrada. Sendo assim aplicou-se na Figura 32 o filtro de Roberts e um nível de threshold e o algoritmo desenvolvido, os resultados foram:

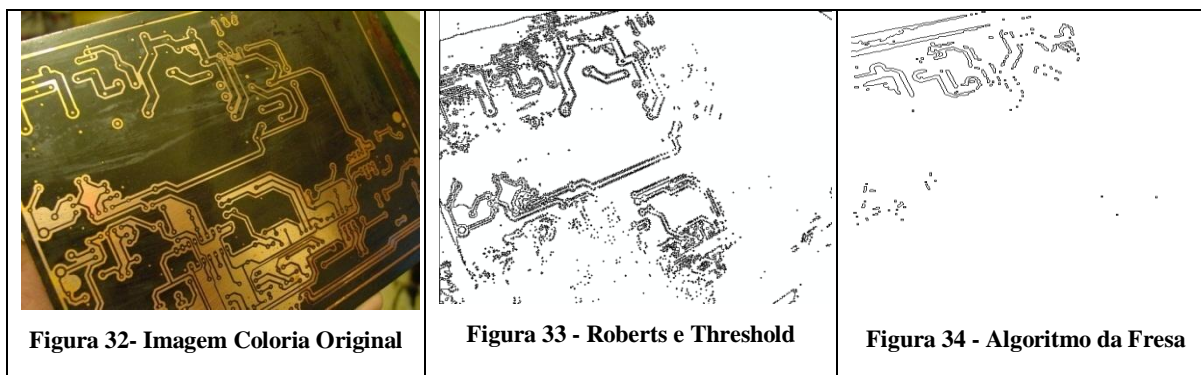


Tabela 17 - Comparação dos Filtros de Detecção de Bordas para uma Imagem Colorida

Com esses resultados podemos tirar algumas conclusões. Estas estão na próxima seção.

4.2.2 Conclusões sobre o algoritmo

Analisando a Tabela 15 pode-se concluir que não há muita diferença entre os diversos filtros aplicados, sendo assim pras análises seguintes foi utilizado apenas um filtro (Roberts). Já em comparação com o algoritmo percebemos grandes diferenças, o principal ponto a se destacar é que o algoritmo é preocupado com a continuidade dos contornos, uma vez que esses serão utilizados para gerarem as instruções para os motores e é interessante manter a continuidade do contorno para que a placa tenha suas trilhas bem definidas.

Da Tabela 16 podemos inferir que o filtro de Roberts possui algumas características que não são interessantes à fresa. Essa característica pode ser vista em detalhe na Figura 35.



Figura 35 - Filtro de Roberts para Circuitos Eletrônicos

Percebe-se que o filtro resulta em contornos com mais de um pixel de largura, isto é, em termos práticos isso obrigaria a fresa a passar duas vezes por esse caminho para gravar o caminho correspondente aos pixels do contorno. Já o algoritmo desenvolvido tem a preocupação de redesenhar as imagens de forma que a borda tenha apenas 1 pixel, e que esses estejam ligados para

impedir a fuga da corrente elétrica. Por isso o algoritmo desenvolvido se mostra mais eficaz para os fins desejados.

4.3 Consumo de energia

Os resultados, segundo a seção 3.2.3 estão dispostos nas tabelas abaixo:

Trecho 1		
Velocidade	Corrente (A)	Potência(W)
1	1.82E-01	2.178
2	0.3055	3.666
3	0.4505	5.406
4	0.62	7.44
5	0.785	9.42

Tabela 18 - Resultados do trecho 1

Trecho 2		
Velocidade	Corrente(A)	Potência(W)
1	1.72E-01	2.058
2	0.321	3.852
3	0.4605	5.526
4	0.605	7.26
5	0.685	8.22

Tabela 19 - Resultados do trecho 2

Trecho 3		
Velocidade	Corrente(A)	Potência(W)
1	3.52E-01	4.224
2	0.698	8.376
3	0.981	11.772
4	1.23	14.76
5	1.43	17.16

Tabela 20 - Resultados do trecho 3

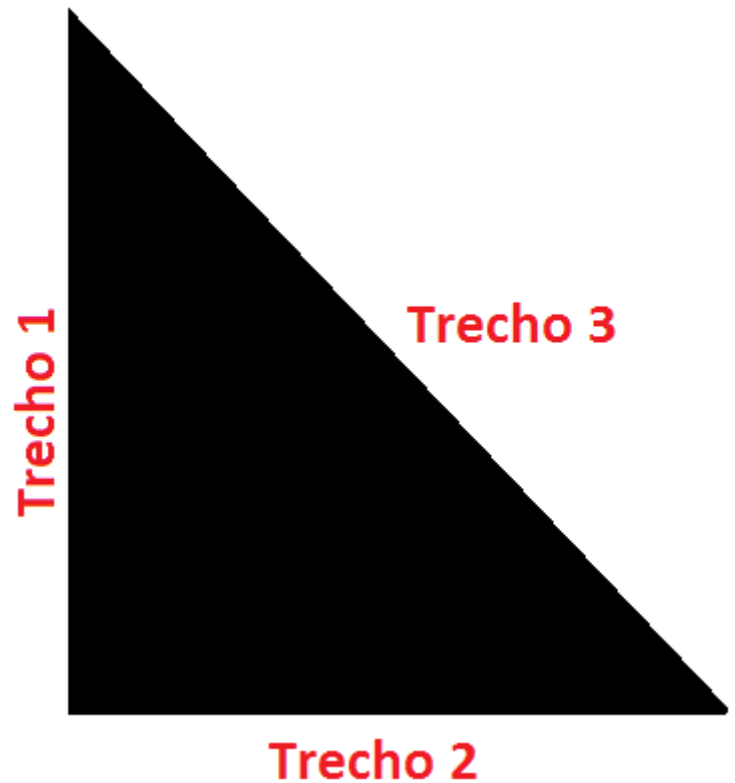


Figura 36- Retângulo para inferir a potência

O sistema não possui um grande consumo, em comparação com as tecnologias presentes, porém deve-se notar que não há o cabeçote fresador, portanto para uma versão final deveria considerar-se a presença de mais um motor (responsável pelo deslocamento em Z) e outro motor, esse sim de alta rotação, responsável pelo giro da broca. Com a adição desses outros dois motores a potência crescerá consideravelmente.

4.4 Conclusões Gerais

O projeto atende aos objetivos, os testes mostraram que a fresa atende ao público amador (principal alvo) porém possui suas limitações. Dentre elas a resolução que impossibilita uma fresagem mais fina (para placas de microeletrônica por exemplo).

Um diferencial do processo é a integração do processo de visão computacional que buscou o melhor algoritmo para as necessidades, sendo assim, através do programa próprio o sistema foge da plataforma tradicional (instruções em G-Code) e traz uma solução simples e acessível com a interface Windows.

O preço também ficou dentro das expectativas. Em comparação com o que há no mercado podemos inferir através dos resultados que existe um bom custo benefício para o projeto.

A interdisciplinaridade foi alcançada. As diversas disciplinas envolvidas, bem como o estágio obrigatório, foram contemplados.

Portanto, considerando o caráter acadêmico, percebe-se claramente que os objetivos foram alcançados, uma vez que as principais funcionalidades do sistema foram alcançadas, isto é, consegue-se completar o ciclo desde a imagem de um arquivo no PC até os movimentos da fresa.

4.5 Trabalhos Futuros

O projeto abre possibilidades para inúmeras melhorias e possui algumas necessidades. Logo de princípio deve-se desenvolver o módulo do cabeçote fresador que devido a problemas de tempo de implementação e financeiros não foi possível ser desenvolvido.

O algoritmo está pronto para receber a implementação do cabeçote fresador e conseqüentemente mais uma dimensão de fresagem (eixo-Z), acrescentar esse novo eixo não requer nenhuma mudança no código já implementado.

Pode-se também implementar sensores de fim de curso, abrindo a porta para o feedback do sistema, daí então talvez fosse necessário implementar um meio de comunicação mais sofisticado podendo ser implantado a comunicação USB.

O código (escrito em VB6) pode ser atualizado para Visual Basic 10, que foi lançado no ano de 2010 e possui uma codificação diferente.

Finalmente, com a fresa concluída, podem-se viabilizar outros fins ligados a Arquitetura, Engenharia Civil ou Agronomia, servindo de auxílio na confecção de maquetes na representação de terrenos com relevos.

Referências Bibliográficas

BOYLESTAD, R. L. **Dispositivos eletrônicos e teoria de circuitos**. 8 Edição. ed. [S.l.]: Pearson Educations do Brasil, 2004.

CNC MÁQUINAS. Produtos. **CNC Máquinas**, 2009. Disponível em: <<http://www.cncmaquinas.com.br/i10/produtoInterna.php?menu=router>>. Acesso em: 25 Setembro 2010.

GONZAGA, A. **Notas de SEL 0339 - INTRODUÇÃO À VISÃO COMPUTACIONAL**. São Carlos: [s.n.], 2010.

IEHK LASER AND CNC SYSTEMS CO. LTD. IEC3030 Desktop/PCB CNC Router. **IEHK LASER AND CNC Systems CO. LTD.**, 2009. Disponível em: <<http://www.iehk.net/Products/CNCIEC3030.html>>. Acesso em: 25 Setembro 2010.

LPKF LASER & ELECTRONICS AG. Products. **LPKF Laser & Electronics AG**, 2010. Disponível em: <<http://www.lpkf.com/products/rapid-pcb-prototyping/circuit-board-plotter/protomat-s100.htm>>. Acesso em: 25 Setembro 2010.

MANFÉ, G. **Desenho Técnico Mecânico: Curso Completo**. [S.l.]: [s.n.], v. 1, 1989.

MESSIAS, A. R. P. Paralela. **RogerCom**, 2006. Disponível em: <<http://www.rogercom.com/pparalela/introducao.htm#modelospp>>. Acesso em: 26 setembro 2010.

MICROSOFT CORPORATION. What's New in Visual Basic 2010. **MSDN**, 2010. Disponível em: <<http://msdn.microsoft.com/en-us/library/ee336123.aspx>>. Acesso em: 25 setembro 2010.

PATNAIK, P. **Handbook of Inorganic Chemicals**. 1. ed. [S.l.]: McGraw-Hill Professional, 2007.

SÉRGIO, R. **A Fresadora**. 1. ed. Rio de Janeiro: Delmar Publisher, 1967.

WILDI, T. **Electrical Machines, Drives, and Power Systems**. 2. ed. [S.l.]: Prentice Hall, 1991.

Anexo 1 – Linguagem Matlab

O texto foi retirado das notas de aula (GONZAGA, 2010).

Detecção de Bordas.

O Toolbox de Processamento de Imagens do MatLab possui uma função que realiza a detecção de bordas. A sintaxe desta função é:

```
[ g, t ] = edge(f, 'method', parameters)
```

Onde f é a Imagem de entrada, g é a Imagem de saída (arranjo lógico com 1 nas posições de bordas detectadas e 0 nas demais) e 'method' corresponde ao tipo de detector utilizado, conforme mostra a Tabela 1. O valor de t é opcional e corresponde ao Threshold que a função utilizará para definir uma borda.

Tabela 1- Detectores de Borda da função *edge*

Edge Detector	Basic Properties
Sobel	Finds edges using the Sobel approximation to the derivatives shown in Fig. 10.5(b).
Prewitt	Finds edges using the Prewitt approximation to the derivatives shown in Fig. 10.5(c).
Roberts	Finds edges using the Roberts approximation to the derivatives shown in Fig. 10.5(d).
Laplacian of a Gaussian (LoG)	Finds edges by looking for zero crossings after filtering $f(x, y)$ with a Gaussian filter.
Zero crossings	Finds edges by looking for zero crossings after filtering $f(x, y)$ with a user-specified filter.
Canny	Finds edges by looking for local maxima of the gradient of $f(x, y)$. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. Therefore, this method is more likely to detect true weak edges.

A sintaxe para o Detector de Sobel usando a função *edge* é:

```
[ g, t ] = edge(f, 'sobel', T, dir)
```

onde T é um Threshold especificado e dir dá a direção preferencial das bordas detectadas (vertical, horizontal, ou both (default)). Se T é especificado então $t = T$. Se T não for especificado, a função `edge` usa um Threshold automático e retorna seu valor em t .

As sintaxes da função `edge`, para os Detectores de Roberts e de Prewitt, são idênticas à sintaxe do Detector de Sobel.

O detector de bordas LoG (Laplaciano da Gaussiana) tem a seguinte sintaxe usando a função `edge`:

$$[g, t] = \text{edge}(f, 'log', T, \text{sigma})$$

onde sigma é o desvio padrão relacionado à suavização realizada pela Gaussiana. O valor default para sigma é 2. T é um Threshold especificado. Fazer $T = 0$ produz bordas em contornos fechados, característica da metodologia LoG.

A sintaxe para o Detector de Bordas de Canny através da função `edge`, é:

$$[g, t] = \text{edge}(f, 'canny', T, \text{sigma})$$

onde T é um vetor $T = [T1, T2]$ contendo os dois valores de Threshold e sigma é o desvio padrão do filtro de suavização. O default para sigma é 1.

Apêndice A – Código do arquivo Paralela.bas

```
Public FioX As Long, FioY As Long

Public Const PDados = 888
Public Const PControle = 890

Declare Sub Out32 Lib "inpout32.dll" (ByVal PortAddr As Integer, ByVal Datum As Integer)
Declare Function Inp32 Lib "inpout32.dll" (ByVal PortAddr As Integer) As Integer
Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Sub PassoP(DXf As Long, DYf As Long, DZf As Long)
    DX = DXf
    DY = DYf
    DZ = DZf
    Dmax = Abs(DX)
    If Abs(DY) > Dmax Then Dmax = Abs(DY)
    If Abs(DZ) > Dmax Then Dmax = Abs(DZ)

    For ContMax = 0 To Dmax

        If DY > 0 Then
            If FioY = 1 Then
                FioY = 3
            ElseIf FioY = 3 Then
                FioY = 2
            ElseIf FioY = 2 Then
                FioY = 6
            ElseIf FioY = 6 Then
                FioY = 4
            ElseIf FioY = 4 Then
                FioY = 12
            End If
        End If
    Next ContMax
End Sub
```

ElseIf FioY = 12 Then

FioY = 8

ElseIf FioY = 8 Then

FioY = 9

ElseIf FioY = 9 Then

FioY = 1

End If

XP = XP + 1

DY = DY - 1

End If

If DY < 0 Then

If FioY = 1 Then

FioY = 9

ElseIf FioY = 9 Then

FioY = 8

ElseIf FioY = 8 Then

FioY = 12

ElseIf FioY = 12 Then

FioY = 4

ElseIf FioY = 4 Then

FioY = 6

ElseIf FioY = 6 Then

FioY = 2

ElseIf FioY = 2 Then

FioY = 3

ElseIf FioY = 3 Then

FioY = 1

End If

XP = XP - 1

DY = DY + 1

End If

```
If DX > 0 Then
  If FioX = 16 Then
    FioX = 48
  ElseIf FioX = 48 Then
    FioX = 32
  ElseIf FioX = 32 Then
    FioX = 96
  ElseIf FioX = 96 Then
    FioX = 64
  ElseIf FioX = 64 Then
    FioX = 192
  ElseIf FioX = 192 Then
    FioX = 128
  ElseIf FioX = 128 Then
    FioX = 144
  ElseIf FioX = 144 Then
    FioX = 16
  End If
```

YP = YP + 1

DX = DX - 1

End If

```
If DX < 0 Then
  If FioX = 16 Then
    FioX = 144
  ElseIf FioX = 144 Then
    FioX = 128
  ElseIf FioX = 128 Then
    FioX = 192
  ElseIf FioX = 192 Then
    FioX = 64
  ElseIf FioX = 64 Then
```

```
FioX = 96
ElseIf FioX = 96 Then
    FioX = 32
ElseIf FioX = 32 Then
    FioX = 48
ElseIf FioX = 48 Then
    FioX = 16
End If

YP = YP - 1
DX = DX + 1
End If

Contador = 0
Do While Contador < Tempo
    If Teste = 0 Then Call Out32(PDatos, FioX + FioY)
    Contador = Contador + 1
Loop

Next ContMax

End Sub
```

Apêndice B – Código do arquivo Variáveis.bas

```
'Fator de escala entre posição em pixels e milímetros
Public Const FatEsc = 386#

'Limite do contador para interrupções periódicas
Public Const LimitePassosTimer = 10

'Número de passos correspondente a um pixel para a reprodução em tamanho real
Public Const Kprop = 100

'Identificador de teste
Public Teste As Integer

'Matrizes para manipulação de imagens
Public Original() As Byte, Auxiliar1() As Byte, Auxiliar2() As Byte

'Contador para interrupções periódicas
Public PixelsDesenhados As Long

'Variável de controle da velocidade da fresa
Public NumeroRepeticoes As Long

'Variáveis para varreduras em loops
Public x As Long, y As Long, Z As Long

'Variável de identificação da próxima posição da fresa
Public XC As Long, YC As Long, ZC As Long

'Variável de identificação da posição anterior da fresa
Public XCA As Long, YCA As Long, ZCA As Long

'Variável de identificação do pixel inicial de um traço contínuo
Public XI As Long, YI As Long, ZI As Long

Public XCant As Integer, YCant As Integer, ZCant As Integer

'Variáveis de deslocamento para o posicionamento inicial
Public PassoX As Long, PassoY As Long, PassoZ As Long

'Registrar posição da fresa em pixels
Public XP As Double, YP As Double, ZP As Double

'Variável de controle da interrupção do rastreador
Public InterrupcaoRastreador As Integer
```

'Variáveis e estruturas para o carregamento da imagem

Private Type SAFEARRAYBOUND

 cElements As Long

 lLbound As Long

End Type

Private Type SAFEARRAY2D

 cDims As Integer

 fFeatures As Integer

 cbElements As Long

 cLocks As Long

 pvData As Long

 Bounds(0 To 1) As SAFEARRAYBOUND

End Type

Private Type BITMAP

 bmType As Long

 bmWidth As Long

 bmHeight As Long

 bmWidthBytes As Long

 bmPlanes As Integer

 bmBitsPixel As Integer

 bmBits As Long

End Type

Public sa1 As SAFEARRAY2D, bmp1 As BITMAP

Public c As New GCommonDialog

Public bContinue As Boolean

Public sFile As String

Apêndice C – Código do arquivo fresadora.frm

```
Private Declare Function GetObjectAPI Lib "gdi32" Alias "GetObjectA" (ByVal hObject As Long, ByVal nCount As Long, lpObject As Any) As Long
Private Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" (pDst As Any, pSrc As Any, ByVal ByteLen As Long)
Private Declare Function VarPtrArray Lib "msvbvm60.dll" Alias "VarPtr" (Ptr() As Any) As Long

'*****
'*****
'*****
'*****
'
'                               Função Principal e de Abrir Arquivo
'*****
'*****
'*****
'*****

Public Sub Form_Load()
    FioX = 16
    FioY = 1
    XP = 0
    YP = 0
    ZP = 0
    NumeroRepeticoes = 400

    TimerPosicionamento.Interval = 1
    TimerFresa.Interval = 1

    CoordX.Caption = "0 mm"
    CoordY.Caption = "0 mm"
    CoordZ.Caption = "0 mm"
    TempoPasso.Caption = 3

    ConfiguracoesIniciais

End Sub

Public Sub ConfiguracoesIniciais()
    Call Out32(PDados, 0)
    InterrupcaoRastreador = 0

    TimerFresa.Enabled = False

    EsquemaOriginal.Visible = False
```

```

vscScroll.Visible = False
hscScroll.Visible = False

Fresar.Enabled = False
Pausar.Enabled = False
BotaoProcessar.Enabled = False
BotaoCarregar.Enabled = False

BotaoPararEsp_Click

End Sub

Public Sub CarregarImagem()
    EsquemaOriginal.Visible = True
    sFile = ""
    bContinue = False
    If (c.VBGetOpenFileName(sFile, , , , , "Arquivos Bitmap (*.BMP)|*.BMP", 1, , ,
"BMP", Principal.hWnd)) Then bContinue = True

    If (bContinue) Then
        EsquemaOriginal = LoadPicture(sFile)
        GetObjectAPI EsquemaOriginal.Picture, Len(bmp1), bmp1

        With sa1
            .cbElements = 1
            .cDims = 2
            .Bounds(0).lLbound = 0
            .Bounds(0).cElements = bmp1.bmHeight
            .Bounds(1).lLbound = 0
            .Bounds(1).cElements = bmp1.bmWidthBytes
            .pvData = bmp1.bmBits
        End With
        EsquemaOriginal.Refresh
        picScrollBar_Resize
        CopyMemory ByVal VarPtrArray(Original), VarPtr(sa1), 4

        If bmp1.bmBitsPixel = 24 Then
            BotaoProcessar.Enabled = True
            Fresar.Enabled = False
            Pausar.Enabled = False
            Status.Caption = "Execute a detecção de bordas se este for o arquivo desejado."
        Else
            Status.Caption = "Formato incompatível. Converta o arquivo para formato bitmap
de 24 bits."
            BotaoProcessar.Enabled = False
        End If
    End Sub

```

End If

End Sub

```
*****  
*****  
*****  
*****  
'  
                Função de Detecção de Borda  
*****  
*****  
*****  
*****
```

Public Sub DetectarBordas()

 BotaoProcessar.Enabled = False

 ReDim Auxiliar1(UBound(Original, 1) / 3, UBound(Original, 2))

 ReDim Auxiliar2(UBound(Original, 1) / 3, UBound(Original, 2))

 For x = 0 To UBound(Original, 1) - 3 Step 3

 For y = 0 To UBound(Original, 2)

 If Original(x, y) < 128 Then

 Original(x, y) = 0

 Else

 Original(x, y) = 255

 End If

 If x <= 3 Or x >= UBound(Original, 1) - 2 * 3 Or y <= 1 Or y >=

UBound(Original, 2) - 1 Then Original(x, y) = 255

 Auxiliar1(x / 3, y) = 255

 Auxiliar2(x / 3, y) = 255

 Next y

 Next x

 For x = 3 To UBound(Original, 1) - 2 * 3 Step 3

 For y = 1 To UBound(Original, 2) - 1

 If Original(x, y) = 0 Then

 If Original(x + 0, y - 1) = 255 Then

 ElseIf Original(x + 0, y + 1) = 255 Then

 ElseIf Original(x - 3, y - 1) = 255 Then

 ElseIf Original(x - 3, y + 0) = 255 Then

 ElseIf Original(x - 3, y + 1) = 255 Then

 ElseIf Original(x + 3, y - 1) = 255 Then

 ElseIf Original(x + 3, y + 0) = 255 Then

 ElseIf Original(x + 3, y + 1) = 255 Then

 Else

 Auxiliar1(x / 3, y) = 0

```

        End If
    End If
Next y
Next x

For x = 0 To UBound(Original, 1) - 3 Step 3
    For y = 0 To UBound(Original, 2)
        Original(x, y) = Auxiliar1(x / 3, y)
        Original(x + 1, y) = Auxiliar1(x / 3, y)
        Original(x + 2, y) = Auxiliar1(x / 3, y)
    Next y
Next x
EsquemaOriginal.Refresh

For x = 3 To UBound(Original, 1) - 2 * 3 Step 3
    For y = 1 To UBound(Original, 2) - 1
        If Auxiliar1(x / 3, y) = 255 Then
            If Auxiliar1(x / 3 - 1, y) = 0 Then Auxiliar2(x / 3, y) = 0
            If Auxiliar1(x / 3 + 0, y - 1) = 0 Then Auxiliar2(x / 3, y) = 0
            If Auxiliar1(x / 3 + 0, y + 1) = 0 Then Auxiliar2(x / 3, y) = 0
            If Auxiliar1(x / 3 + 1, y) = 0 Then Auxiliar2(x / 3, y) = 0
        End If
    Next y
Next x

For x = 0 To UBound(Original, 1) - 3 Step 3
    For y = 0 To UBound(Original, 2)
        Original(x, y) = Auxiliar2(x / 3, y)
        Original(x + 1, y) = Auxiliar2(x / 3, y)
        Original(x + 2, y) = Auxiliar2(x / 3, y)
    Next y
Next x
EsquemaOriginal.Refresh

Fresar.Enabled = True
Pausar.Enabled = False
Status.Caption = "Inicie a fresagem se a detecção foi satisfatória. Caso contrário, clique
em Novo."

x = 0
y = 0
Z = 0

End Sub

'*****
*****

```

```

*****
*****
'
          Funções dos Botões de Posicionamento Inicial
*****
*****
!*****
*****

Public Sub BotaoDefOrigem_Click()
    x = 0
    y = 0
    Z = 0
    XC = 0
    YC = 0
    ZC = 0
    XCA = 0
    YCA = 0
    ZCA = 0
    XI = 0
    YI = 0
    ZI = 0
    XP = 0
    YP = 0
    ZP = 0
    PassoX = 0
    PassoY = 0
    PassoZ = 0

    BotaoCarregar.Enabled = True
    BotaoDefOrigem.Enabled = False
    BotaoMovimentarXN.Enabled = False
    BotaoMovimentarXP.Enabled = False
    BotaoMovimentarYN.Enabled = False
    BotaoMovimentarYP.Enabled = False
    BotaoPararEsp.Enabled = False

    CoordX.Caption = "0 mm"
    CoordY.Caption = "0 mm"
    CoordZ.Caption = "0 mm"
    Status.Caption = "Carregue o arquivo com o esquema do circuito impresso."
End Sub

Public Sub BotaoMovimentarXN_Click()
    PassoX = -99
    BotaoMovimentarXN.Enabled = False
    BotaoMovimentarXP.Enabled = True
    BotaoPararEsp.Enabled = True

```

```

    BotaoDefOrigem.Enabled = False
    TimerPosicionamento.Enabled = True
    Status.Caption = "Pare a fresa quando estiver posicionada na origem da placa."
End Sub

Public Sub BotaoMovimentarXP_Click()
    PassoX = 99
    BotaoMovimentarXP.Enabled = False
    BotaoMovimentarXN.Enabled = True
    BotaoPararEsp.Enabled = True
    BotaoDefOrigem.Enabled = False
    TimerPosicionamento.Enabled = True
    Status.Caption = "Pare a fresa quando estiver posicionada na origem da placa."
End Sub

Public Sub BotaoMovimentarYN_Click()
    PassoY = -99
    BotaoMovimentarYN.Enabled = False
    BotaoMovimentarYP.Enabled = True
    BotaoPararEsp.Enabled = True
    BotaoDefOrigem.Enabled = False
    TimerPosicionamento.Enabled = True
    Status.Caption = "Pare a fresa quando estiver posicionada na origem da placa."
End Sub

Public Sub BotaoMovimentarYP_Click()
    PassoY = 99
    BotaoMovimentarYP.Enabled = False
    BotaoMovimentarYN.Enabled = True
    BotaoPararEsp.Enabled = True
    BotaoDefOrigem.Enabled = False
    TimerPosicionamento.Enabled = True
    Status.Caption = "Pare a fresa quando estiver posicionada na origem da placa."
End Sub

Public Sub BotaoPararEsp_Click()
    Call Out32(PDados, 0)

    PassoX = 0
    PassoY = 0
    PassoZ = 0
    TimerPosicionamento.Enabled = False
    BotaoPararEsp.Enabled = False
    BotaoDefOrigem.Enabled = True

    BotaoMovimentarXN.Enabled = True
    BotaoMovimentarXP.Enabled = True

```

```
BotaoMovimentarYN.Enabled = True
BotaoMovimentarYP.Enabled = True
Status.Caption = "Confirme o ponto de origem se o posicionamento inicial estiver
concluído."
End Sub
```

```
*****
*****
'*****
*****
'
                Funções dos Botões de Controle do Processo
'*****
*****
'*****
*****
```

```
Public Sub Fresar_Click()
    MenuNovo.Enabled = False
    Fresar.Enabled = False
    Pausar.Enabled = True
    MenuSair.Enabled = False
    BotaoCarregar.Enabled = False
    TimerFresa.Enabled = True
    Status.Caption = "Fresando. É possível pausar o processo durante a fresagem dos
contornos."
End Sub
```

```
Public Sub Pausar_Click()
    MenuNovo.Enabled = True
    TimerFresa.Enabled = False
    Fresar.Enabled = True
    Pausar.Enabled = False
    MenuSair.Enabled = True
    Status.Caption = "Fresagem pausada. Clique em Fresar para continuar, ou em Sair para
interromper."
End Sub
```

```
*****
*****
'*****
*****
'
                Funções dos Botões de Imagem e Menus
'*****
*****
'*****
*****
```

```
Public Sub BotaoProcessar_Click()
    DetectarBordas
End Sub
```

```
Public Sub BotaoCarregar_Click()
    CarregarImagem
End Sub
```

```
Private Sub MenuNovo_Click()
    Form_Load
End Sub
```

```
Public Sub MenuSair_Click()
    Call Out32(PDados, 0)
    Unload Principal
End
End Sub
```

```
*****
*****
*****
*****
'
                Funções dos Botões de Velocidade
*****
*****
*****
*****
```

```
Public Sub DiminuirVelocidade_Click()
    AumentarVelocidade.Enabled = True
    If NumeroRepeticoes = 250 Then
        TempoPasso.Caption = 4
        NumeroRepeticoes = 300
    ElseIf NumeroRepeticoes = 300 Then
        TempoPasso.Caption = 3
        NumeroRepeticoes = 400
    ElseIf NumeroRepeticoes = 400 Then
        TempoPasso.Caption = 2
        NumeroRepeticoes = 600
    ElseIf NumeroRepeticoes = 600 Then
        TempoPasso.Caption = 1
        NumeroRepeticoes = 1000
        DiminuirVelocidade.Enabled = False
    End If
End Sub
```

```
Public Sub AumentarVelocidade_Click()
```

```

DiminuirVelocidade.Enabled = True
If NumeroRepeticoes = 1000 Then
    TempoPasso.Caption = 2
    NumeroRepeticoes = 600
ElseIf NumeroRepeticoes = 600 Then
    TempoPasso.Caption = 3
    NumeroRepeticoes = 400
ElseIf NumeroRepeticoes = 400 Then
    TempoPasso.Caption = 4
    NumeroRepeticoes = 300
ElseIf NumeroRepeticoes = 300 Then
    TempoPasso.Caption = 5
    NumeroRepeticoes = 250
    AumentarVelocidade.Enabled = False
End If
End Sub

'*****
'*****
'*****
'*****
'
'                               Funções dos Botões de Controle do Processo
'*****
'*****
'*****
'*****

Public Sub hscScroll_Scroll()
    hscScroll_Change
End Sub

Public Sub hscScroll_Change()
    EsquemaOriginal.Left = -Screen.TwipsPerPixelY * hscScroll.Value
End Sub

Public Sub vscScroll_Change()
    EsquemaOriginal.Top = -Screen.TwipsPerPixelY * vscScroll.Value
End Sub

Public Sub vscScroll_Scroll()
    vscScroll_Change
End Sub

Public Sub picScrollBar_Resize()
    On Error Resume Next
    hscScroll.Visible = (picScrollBar.ScaleWidth - vscScroll.Width <
EsquemaOriginal.Width)

```

```

    vscScroll.Visible = (picScrollBar.ScaleHeight - hscScroll.Height <
EsquemaOriginal.Height)
    If (hscScroll.Visible) Then
        hscScroll.Max = (EsquemaOriginal.Width - picScrollBar.ScaleWidth +
vscScroll.Width * Abs(vscScroll.Visible)) \ Screen.TwipsPerPixelX
        hscScroll.SmallChange = 32
    End If
    If (vscScroll.Visible) Then
        vscScroll.Max = (EsquemaOriginal.Height - picScrollBar.ScaleHeight +
hscScroll.Height * Abs(hscScroll.Visible)) \ Screen.TwipsPerPixelY
        vscScroll.SmallChange = 32
    End If
End Sub

'*****
'*****
'*****
'*****
'
'           Timers de Controle
'*****
'*****
'*****
'*****

Public Sub TimerPosicionamento_Timer()
    Call PassoP(PassoX, PassoY, PassoZ)
    CoordX.Caption = Format(XP / FatEsc, "#0.00") & " mm"
    CoordY.Caption = Format(YP / FatEsc, "#0.00") & " mm"
End Sub

Public Sub TimerFresa_Timer()
    CoordX.Caption = Format(XP / FatEsc, "#0.00") & " mm"
    CoordY.Caption = Format(YP / FatEsc, "#0.00") & " mm"
    ProcessarCNC
End Sub

'*****
'*****
'*****
'*****
'
'           Funções de Controle do CNC
'*****
'*****
'*****
'*****

Public Sub Final()

```

```

If Teste = 0 Then Call Out32(PDados, 0)
TimerFresa.Enabled = False

BotaoMovimentarXN.Enabled = True
BotaoMovimentarXP.Enabled = True
BotaoMovimentarYN.Enabled = True
BotaoMovimentarYP.Enabled = True
BotaoPararEsp.Enabled = False
BotaoDefOrigem.Enabled = True

Fresar.Enabled = False
Pausar.Enabled = False
MenuSair.Enabled = True
MenuNovo.Enabled = True
End Sub

Public Sub ProcessarCNC()
    Fim = 1
    Do While x <= UBound(Original, 1) - 3
        Fim = 0
        If y = UBound(Original, 2) + 1 Then y = 0
        Do While y <= UBound(Original, 2)
            If Auxiliar2(x / 3, y) = 0 Or InterrupcaoRastreador = 1 Then
                If InterrupcaoRastreador = 0 Then
                    XI = x / 3
                    YI = y
                    Call Movimentar(x / 3, y, 0, 0)
                    Auxiliar2(XC, YC) = 128
                End If
                Rastreador
                If XC * YC > 0 Then EsquemaOriginal.Refresh
            End If
            If InterrupcaoRastreador = 1 Then Exit Do
            y = y + 1
        Loop
        If InterrupcaoRastreador = 1 Then Exit Do
        x = x + 3
    Loop
    If Fim = 1 Then Final
End Sub

Public Sub Movimentar(XX As Long, YY As Long, ZZ As Long, Cor As Integer)
    XCA = XC
    YCA = YC
    ZCA = ZC
    XC = XX
    YC = YY

```

```

ZC = ZZ
Call PassoP(Kprop * (XC - XCA), Kprop * (YC - YCA), Kprop * (ZC - ZCA))
If Cor = 1 Then
    Auxiliar2(XC, YC) = 128
    Original(3 * XC, YC) = 0
    Original(3 * XC + 1, YC) = 0
    Original(3 * XC + 2, YC) = 255
End If
End Sub

Public Sub Rastreador()
    PixelsDesenhados = 0
    InterrupcaoRastreador = 0

    Do While 1
        If PixelsDesenhados = LimitePassosTimer Then
            InterrupcaoRastreador = 1
            Exit Do
        End If
        PixelsDesenhados = PixelsDesenhados + 1

        If Auxiliar2(XC, YC - 1) = 0 Then
            Call Movimentar(XC, YC - 1, 0, 1)
        ElseIf Auxiliar2(XC + 1, YC) = 0 Then
            Call Movimentar(XC + 1, YC, 0, 1)
        ElseIf Auxiliar2(XC, YC + 1) = 0 Then
            Call Movimentar(XC, YC + 1, 0, 1)
        ElseIf Auxiliar2(XC - 1, YC) = 0 Then
            Call Movimentar(XC - 1, YC, 0, 1)
        ElseIf Auxiliar2(XC - 1, YC - 1) = 0 Then
            Call Movimentar(XC - 1, YC - 1, 0, 1)
        ElseIf Auxiliar2(XC - 1, YC + 1) = 0 Then
            Call Movimentar(XC - 1, YC + 1, 0, 1)
        ElseIf Auxiliar2(XC + 1, YC - 1) = 0 Then
            Call Movimentar(XC + 1, YC - 1, 0, 1)
        ElseIf Auxiliar2(XC + 1, YC + 1) = 0 Then
            Call Movimentar(XC + 1, YC + 1, 0, 1)
        Else
            Call Movimentar(XI, YI, 0, 0)
        End If
        Exit Sub
    Loop
End Sub

```

Apêndice D – Pseudo-Código para entendimento do algoritmo principal

PROCESSO Form_Load

FioX ← 16
FioY ← 1
XP ← 0
YP ← 0
ZP ← 0
NumeroRepeticoes ← 400
TimerPosicionamento.Interval ← 1
TimerFresa.Interval ← 1
CoordX ← "0 mm"
CoordY ← "0 mm"
CoordZ ← "0 mm"
TempoPasso ← 3
Chamar ConfiguracoesIniciais

PROCESSO ConfiguracoesIniciais

Enviar zero à porta paralela
InterrupcaoRastreador ← 0
Desabilitar TimerFresa
Ocultar EsquemaOriginal
Ocultar vscScroll
Ocultar hscScroll
Desabilitar Fresar
Desabilitar Pausar
Desabilitar BotaoProcessar
Desabilitar BotaoCarregar
Chamar BotaoPararEsp_Click

PROCESSO BotaoDefOrigem_Click

XC ← 0

YC ← 0
ZC ← 0
XCA ← 0
YCA ← 0
ZCA ← 0
XI ← 0
YI ← 0
ZI ← 0
XP ← 0
YP ← 0
ZP ← 0
CoordX ← "0 mm"
CoordY ← "0 mm"
CoordZ ← "0 mm"
Status ← "Carregue o arquivo com o esquema do circuito impresso."
Habilitar BotaoCarregar
Desabilitar BotaoDefOrigem
Desabilitar BotaoMovimentarXN
Desabilitar BotaoMovimentarXP
Desabilitar BotaoMovimentarYN
Desabilitar BotaoMovimentarYP
Desabilitar BotaoPararEsp

PROCESSO BotaoMovimentarXN_Click

PassoX ← -99
Desabilitar BotaoMovimentarXN
Habilitar BotaoMovimentarXP
Habilitar BotaoPararEsp
Desabilitar BotaoDefOrigem
Habilitar TimerPosicionamento
Status ← "Pare a fresa quando estiver posicionada na origem da placa."

PROCESSO BotaoMovimentarXP_Click

PassoX ← 99

Desabilitar BotaoMovimentarXP

Habilitar BotaoMovimentarXN

Habilitar BotaoPararEsp

Desabilitar BotaoDefOrigem

Habilitar TimerPosicionamento

Status ← "Pare a fresa quando estiver posicionada na origem da placa."

PROCESSO BotaoMovimentarYN_Click

PassoY ← -99

Desabilitar BotaoMovimentarYN

Habilitar BotaoMovimentarYP

Habilitar BotaoPararEsp

Desabilitar BotaoDefOrigem

Habilitar TimerPosicionamento

Status ← "Pare a fresa quando estiver posicionada na origem da placa."

PROCESSO BotaoMovimentarYP_Click

PassoY ← 99

Desabilitar BotaoMovimentarYP

Habilitar BotaoMovimentarYN

Habilitar BotaoPararEsp

Desabilitar BotaoDefOrigem

Habilitar TimerPosicionamento

Status ← "Pare a fresa quando estiver posicionada na origem da placa."

PROCESSO BotaoPararEsp_Click

Enviar zero à porta paralela

PassoX ← 0

PassoY ← 0

PassoZ ← 0

Desabilita TimerPosicionamento

Desabilita BotaoPararEsp

Habilita BotaoDefOrigem

Habilita BotaoMovimentarXN

Habilita BotaoMovimentarXP

Habilita BotaoMovimentarYN

Habilita BotaoMovimentarYP

Status ← "Confirme o ponto de origem se o posicionamento inicial estiver concluído."

PROCESSO Fresar_Click

Desabilita MenuNovo

Desabilita Fresar

Habilita Pausar

Desabilita MenuSair

Desabilita BotaoCarregar

Habilita TimerFresa

Status ← "Fresando. É possível pausar o processo durante a fresagem dos contornos."

PROCESSO Pausar_Click

Habilita MenuNovo

Desabilita TimerFresa

Habilita Fresar

Desabilita Pausar

Habilita MenuSair

Status ← "Fresagem pausada. Clique em Fresar para continuar, ou em Sair para interromper."

PROCESSO BotaoProcessar_Click

Desabilita BotaoProcessar

Redimensionar Auxiliar1 para receber imagem

Redimensionar Auxiliar2 para receber imagem

```

Para x = 0 Até Limite Horizontal(Original) - 3 Passo 3
  Para y = 0 Até Limite Vertical(Original)
    Se Original(x, y) < 128 Então
      Original(x, y) ← 0
    Senão
      Original(x, y) ← 255
    Fim Se
    Se x <= 3 Ou x >= Limite Horizontal(Original) - 2 * 3 Ou y <= 1 Ou y >= Limite
Vertical(Original) - 1 Então Original(x, y) ← 255
      Auxiliar1(x / 3, y) ← 255
      Auxiliar2(x / 3, y) ← 255
    Fim Para
  Fim Para
Para x = 3 Até Limite Horizontal(Original) - 2 * 3 Passo 3
  Para y = 1 Até Limite Vertical(Original) - 1
    Se Original(x, y) = 0 Então
      Se Original(x + 0, y - 1) = 255 Então
        Senão Se Original(x + 0, y + 1) = 255 Então
          Senão Se Original(x - 3, y - 1) = 255 Então
            Senão Se Original(x - 3, y + 0) = 255 Então
              Senão Se Original(x - 3, y + 1) = 255 Então
                Senão Se Original(x + 3, y - 1) = 255 Então
                  Senão Se Original(x + 3, y + 0) = 255 Então
                    Senão Se Original(x + 3, y + 1) = 255 Então
                      Senão
                        Auxiliar1(x / 3, y) ← 0
                      Fim Se
                    Fim Se
                  Fim Para
                Fim Para
              Fim Para
            Fim Para
          Fim Para
        Fim Para
      Fim Para
    Fim Para
  Para x = 0 Até Limite Horizontal(Original) - 3 Passo 3
    Para y = 0 Até Limite Vertical(Original)
      Original(x, y) ← Auxiliar1(x / 3, y)
      Original(x + 1, y) ← Auxiliar1(x / 3, y)
    Fim Para
  Fim Para

```

```

Original(x + 2, y) ← Auxiliar1(x / 3, y)
Fim Para
Fim Para
Atualizar EsquemaOriginal
Para x = 3 Até Limite Horizontal(Original) - 2 * 3 Passo 3
  Para y = 1 Até Limite Vertical(Original) - 1
    Se Auxiliar1(x / 3, y) = 255 Então
      Se Auxiliar1(x / 3 - 1, y) = 0 Então Auxiliar2(x / 3, y) ← 0
      Se Auxiliar1(x / 3 + 0, y - 1) = 0 Então Auxiliar2(x / 3, y) ← 0
      Se Auxiliar1(x / 3 + 0, y + 1) = 0 Então Auxiliar2(x / 3, y) ← 0
      Se Auxiliar1(x / 3 + 1, y) = 0 Então Auxiliar2(x / 3, y) ← 0
    Fim Se
  Fim Para
Fim Para
Para x = 0 Até Limite Horizontal(Original) - 3 Passo 3
  Para y = 0 Até Limite Vertical(Original)
    Original(x, y) ← Auxiliar2(x / 3, y)
    Original(x + 1, y) ← Auxiliar2(x / 3, y)
    Original(x + 2, y) ← Auxiliar2(x / 3, y)
  Fim Para
Fim Para
Atualizar EsquemaOriginal
Habilitar Fresar
Desabilitar Pausar
Status ← "Inicie a fresagem se a detecção foi satisfatória. Caso contrário, clique em
Novo."
x ← 0
y ← 0

PROCESSO MenuNovo_Click
  Chamar ConfiguracoesIniciais

PROCESSO MenuSair_Click

```

Enviar zero à porta paralela

Fechar Principal

Sair

PROCESSO DiminuirVelocidade_Click

Habilitar AumentarVelocidade

Se NumeroRepeticoes = 250 Então

TempoPasso ← 4

NumeroRepeticoes ← 300

Senão Se NumeroRepeticoes = 300 Então

TempoPasso ← 3

NumeroRepeticoes ← 400

Senão Se NumeroRepeticoes = 400 Então

TempoPasso ← 2

NumeroRepeticoes ← 600

Senão Se NumeroRepeticoes = 600 Então

TempoPasso ← 1

NumeroRepeticoes ← 1000

Desabilitar DiminuirVelocidade

Fim Se

PROCESSO AumentarVelocidade_Click

Habilitar DiminuirVelocidade

Se NumeroRepeticoes = 1000 Então

TempoPasso ← 2

NumeroRepeticoes ← 600

Senão Se NumeroRepeticoes = 600 Então

TempoPasso ← 3

NumeroRepeticoes ← 400

Senão Se NumeroRepeticoes = 400 Então

TempoPasso ← 4

NumeroRepeticoes ← 300

Senão Se NumeroRepeticoes = 300 Então

TempoPasso ← 5
NumeroRepeticoes ← 250
Desabilita AumentarVelocidade

Fim Se

PROCESSO TimerPosicionamento_Timer

Chamar PassoP(PassoX, PassoY, PassoZ)

CoordX ← XP / FatEsc & " mm"

CoordY ← YP / FatEsc & " mm"

PROCESSO TimerFresa_Timer

Chamar ProcessarCNC

CoordX ← Format(XP / FatEsc, "#0.00") & " mm"

CoordY ← Format(YP / FatEsc, "#0.00") & " mm"

PROCESSO Final

Enviar zero à porta paralela

Desabilita TimerFresa

Habilita BotaoMovimentarXN

Habilita BotaoMovimentarXP

Habilita BotaoMovimentarYN

Habilita BotaoMovimentarYP

Desabilita BotaoPararEsp

Habilita BotaoDefOrigem

Desabilita Fresar

Desabilita Pausar

Habilita MenuSair

Habilita MenuNovo

PROCESSO ProcessarCNC

Fim ← 1

```

Enquanto x <= Limite Horizontal(Original) - 3
  Fim ← 0
  Se y = Limite Vertical(Original) + 1 Então y ← 0
  Enquanto y <= Limite Vertical(Original)
    Se Auxiliar2(x / 3, y) = 0 Ou InterrupcaoRastreador = 1 Então
      Se InterrupcaoRastreador = 0 Então
        XI ← x / 3
        YI ← y
        Chamar Movimentar(x / 3, y, 0, 0)
        Auxiliar2(XC, YC) ← 128
      Fim Se
      Chamar Rastreador
      Se XC + YC > 0 Então Atualizar EsquemaOriginal
    Fim Se
    Se InterrupcaoRastreador = 1 Então Exit Do
    y ← y + 1
  Fim Enquanto
  Se InterrupcaoRastreador = 1 Então Exit Do
  x ← x + 3
Fim Enquanto
Se Fim = 1 Então Final

```

PROCESSO Movimentar(XX Como Long, YY Como Long, ZZ Como Long, Cor Como Integer)

```

  XCA ← XC
  YCA ← YC
  ZCA ← ZC
  XC ← XX
  YC ← YY
  ZC ← ZZ
  Chamar PassoP(Kprop * (XC - XCA), Kprop * (YC - YCA), Kprop * (ZC - ZCA))
  Se Cor = 1 Então
    Auxiliar2(XC, YC) ← 128

```

```
Original(3 * XC, YC) ← 0
Original(3 * XC + 1, YC) ← 0
Original(3 * XC + 2, YC) ← 255
```

Fim Se

PROCESSO Rastreador

```
PixelsDesenhados ← 0
InterrupcaoRastreador ← 0
```

Enquanto 1

Se PixelsDesenhados = LimitePassosTimer Então

InterrupcaoRastreador ← 1

Exit Do

Fim Se

PixelsDesenhados ← PixelsDesenhados + 1

Se Auxiliar2(XC, YC - 1) = 0 Então

Chamar Movimentar(XC, YC - 1, 0, 1)

Senão Se Auxiliar2(XC + 1, YC) = 0 Então

Chamar Movimentar(XC + 1, YC, 0, 1)

Senão Se Auxiliar2(XC, YC + 1) = 0 Então

Chamar Movimentar(XC, YC + 1, 0, 1)

Senão Se Auxiliar2(XC - 1, YC) = 0 Então

Chamar Movimentar(XC - 1, YC, 0, 1)

Senão Se Auxiliar2(XC - 1, YC - 1) = 0 Então

Chamar Movimentar(XC - 1, YC - 1, 0, 1)

Senão Se Auxiliar2(XC - 1, YC + 1) = 0 Então

Chamar Movimentar(XC - 1, YC + 1, 0, 1)

Senão Se Auxiliar2(XC + 1, YC - 1) = 0 Então

Chamar Movimentar(XC + 1, YC - 1, 0, 1)

Senão Se Auxiliar2(XC + 1, YC + 1) = 0 Então

Chamar Movimentar(XC + 1, YC + 1, 0, 1)

Senão

Chamar Movimentar(XI, YI, 0, 0)

Exit Sub

Fim Se

Fim Enquanto

PROCESSO PassoP(DXf Como Long, DYf Como Long, DZf Como Long)

DX ← DXf

DY ← DYf

DZ ← DZf

Dmax ← Módulo(DX)

Se Módulo (DY) > Dmax Então Dmax ← Módulo(DY)

Se Módulo (DZ) > Dmax Então Dmax ← Módulo(DZ)

Para ContMax = 0 Até Dmax

Se DY > 0 Então

Se FioY = 1 Então

FioY ← 3

Senão Se FioY = 3 Então

FioY ← 2

Senão Se FioY = 2 Então

FioY ← 6

Senão Se FioY = 6 Então

FioY ← 4

Senão Se FioY = 4 Então

FioY ← 12

Senão Se FioY = 12 Então

FioY ← 8

Senão Se FioY = 8 Então

FioY ← 9

Senão Se FioY = 9 Então

FioY ← 1

Fim Se

XP ← XP + 1

DY ← DY - 1

Fim Se

Se $DY < 0$ Então

Se $FioY = 1$ Então

$FioY \leftarrow 9$

Senão Se $FioY = 9$ Então

$FioY \leftarrow 8$

Senão Se $FioY = 8$ Então

$FioY \leftarrow 12$

Senão Se $FioY = 12$ Então

$FioY \leftarrow 4$

Senão Se $FioY = 4$ Então

$FioY \leftarrow 6$

Senão Se $FioY = 6$ Então

$FioY \leftarrow 2$

Senão Se $FioY = 2$ Então

$FioY \leftarrow 3$

Senão Se $FioY = 3$ Então

$FioY \leftarrow 1$

Fim Se

$XP \leftarrow XP - 1$

$DY \leftarrow DY + 1$

Fim Se

Se $DX > 0$ Então

Se $FioX = 16$ Então

$FioX \leftarrow 48$

Senão Se $FioX = 48$ Então

$FioX \leftarrow 32$

Senão Se $FioX = 32$ Então

$FioX \leftarrow 96$

Senão Se $FioX = 96$ Então

$FioX \leftarrow 64$

Senão Se $FioX = 64$ Então

$FioX \leftarrow 192$

Senão Se $FioX = 192$ Então

```

FioX ← 128
  Senão Se FioX = 128 Então
    FioX ← 144
  Senão Se FioX = 144 Então
    FioX ← 16
  Fim Se
  YP ← YP + 1
  DX ← DX - 1
Fim Se
Se DX < 0 Então
  Se FioX = 16 Então
    FioX ← 144
  Senão Se FioX = 144 Então
    FioX ← 128
  Senão Se FioX = 128 Então
    FioX ← 192
  Senão Se FioX = 192 Então
    FioX ← 64
  Senão Se FioX = 64 Então
    FioX ← 96
  Senão Se FioX = 96 Então
    FioX ← 32
  Senão Se FioX = 32 Então
    FioX ← 48
  Senão Se FioX = 48 Então
    FioX ← 16
  Fim Se
  YP ← YP - 1
  DX ← DX + 1
Fim Se
Contador ← 0
Enquanto Contador < Tempo
  Se Teste = 0 Então Chamar Out32(PDados, FioX + FioY)

```

Contador \leftarrow Contador + 1

Fim Enquanto

Fim Para