

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA MECÂNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA MECÂNICA**

Fábio José Souza

**USINAGEM REMOTA DE PEÇAS PRISMÁTICAS VIA
INTERNET EM UMA MÁQUINA CNC ADERENTE AO
PADRÃO STEP-NC.**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina para obtenção do Grau de Mestre em Engenharia Mecânica.
Orientador: Prof. João Carlos Espíndola Ferreira, Ph.D.
Co-orientador: Prof. Carlos Alberto Martin, Dr. Ing

Florianópolis

2014

Catálogo na fonte elaborada pela biblioteca da
Universidade Federal de Santa Catarina

Souza, Fábio José

Usinagem remota de peças prismáticas via internet em uma máquina CNC aderente ao padrão STEP-NC / Fábio José Souza ; orientador, João Carlos Espíndola Ferreira ; coorientador, Carlos Alberto Martin. - Florianópolis, SC, 2014.

134 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Mecânica.

Inclui referências

1. Engenharia Mecânica. 2. STEP-NC. 3. Código G. 4. Fresadora CNC. 5. Padronização. I. Espíndola Ferreira, João Carlos. II. Martin, Carlos Alberto. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Mecânica. IV. Título.

Fábio José Souza

Usinagem remota de peças prismáticas via internet em uma máquina CNC aderente ao padrão STEP-NC.

Esta dissertação foi julgada e aprovada em 22/08/2014, para a obtenção do Título de Mestre em Engenharia Mecânica, área de concentração em Fabricação, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Mecânica, Departamento de Engenharia Mecânica, Universidade Federal de Santa Catarina.

Prof. Armando Albertazzi Gonçalves Jr., Dr.Eng.
Coordenador do Curso

Prof. João Carlos Espíndola Ferreira, Ph.D.
Orientador
Universidade Federal de Santa Catarina

Prof. Carlos Alberto Martin, Dr.Eng.
Co-Orientador
Universidade Federal de Santa Catarina

Banca Examinadora:

Prof. Roberto Silvio Ubertino Rosso Jr, Ph.D.
Universidade do Estado de Santa Catarina

Prof. Adriano Fagali de Souza, Dr.Eng.
Universidade Federal de Santa Catarina

Prof. Carlos Henrique Ahrens, Dr.Eng.
Universidade Federal de Santa Catarina

Florianópolis, 22 de agosto de 2014.

*A meus pais, Aurélio e Janete
pelo amor incondicional, meus irmãos
Marcos e Monique, meu grande amigo
e cunhado Alexandre pelo incentivo e a
minha noiva e futura esposa Elisângela
pelo amor, apoio e compreensão
dedicados neste período.*

AGRADECIMENTOS

Primeiramente, a Deus, por tudo e pela existência, sem ele eu não teria chegado até aqui.

Agradeço aos meus pais, Aurélio e Janete, aos meus avós maternos Sady e Lucy e aos meus avós paternos Sérgio e Amadora, pelo amor, ensinamentos, princípios, confiança, carinho, preocupação, apoio irrestrito e acima de tudo, por acreditar em todas as decisões que tomei até hoje. A minha irmã Monique por me orientarem na escrita desta dissertação e ao meu amigo Júlio Alexandre Teixeira por sempre me receber no Laboratório de Vibração Acústica (LVA) da UFSC, me aconselhando e orientando dentro do seu alcance.

Aos meus familiares, principalmente a Sady Machado Filho e Jecy Barbosa Gonçalves Junior pelo apoio, suporte e ensinamentos na parte eletrônica deste projeto.

A minha linda noiva Elisangela pelo amor, apoio, paciência e compreensão dedicados neste período para conclusão de mais este diploma na minha vida.

Queria muito agradecer à Universidade Federal de Santa Catarina (UFSC) e ao Programa de Pós-Graduação em Engenharia Mecânica (POSMEC) e principalmente ao CNPq, pela concessão da bolsa durante todo o mestrado.

Ao meu orientador Prof. João Carlos Espíndola Ferreira Ph.D, pela valiosa orientação, por me mostrar esta nova tecnologia que está sendo estudada em nível mundial, pela oportunidade, principalmente pela paciência e por aceitar a orientação.

Meu Co-orientador Prof. Carlos Alberto Martin Dr.Eng e ao Prof. Antônio Dourado Dr. Eng, pela confiança, por acreditarem nesse projeto e cederem o Laboratório de Hardware II para assumir a responsabilidade do local e assim concluir essa pesquisa.

Ao bolsista Wagner Gascho pelos meses dedicados neste projeto, aos amigos de laboratório, Julio César Ticona Benavente e Frank Ajata que sempre que puderam, ajudaram dando sugestões no projeto.

A todos os meus amigos de Florianópolis e aos que fiz em São Paulo, pela força, por entenderem minhas ausências. Por fim, as demais pessoas do meu convívio, que de uma forma ou de outra contribuíram para a realização e conclusão dessa etapa da minha vida.

Muito obrigado!

“Se é pra viver que a vida me leve...”

(Elisangela R. Celli)

RESUMO

O atual padrão de programação dos centros de usinagem é ainda a norma (ISO 6983-1), conhecido como código G, que foi desenvolvido no final de 1950. Essa linguagem contém informações de baixo nível que efetuam o movimento da ferramenta no processo de usinagem, dificultando a tomada de decisões pelo operador no processo de usinagem. Por este motivo, a norma (ISO 6983-1) é uma das atuais limitações de flexibilidade e interoperabilidade na fabricação de peças. Diante disto, nos anos 1990 foi proposto um novo modelo de dados para os processos de usinagem, estendendo o padrão STEP - norma (ISO 10303-1) - para o novo padrão, denominado STEP-NC (ISO 14649-1), que foi desenvolvido como resultado de vários projetos realizados por empresas e universidades. No presente trabalho descreve-se o desenvolvimento e validação de *software* e *hardware* que permite o usuário comandar uma fresadora CNC tipo pórtico de maneira que ela venha a usinar peças de maneira aderente ao padrão STEP-NC norma (ISO 14649-11). O STEP-NC se baseia no conceito de *workingsteps*, *features* de usinagem e operações de usinagem, entre outras entidades descritas pela norma (ISO 14649-1). O *software* STEP-NC Controller foi desenvolvido utilizando a linguagem de programação Java, que possibilitou a criação de uma *interface* de controle da fresadora CNC para o processo de usinagem. Foi utilizada a programação do código aberto GRBL (*software* de controle e movimentos em máquinas) aderente à norma RS-274 (códigos G e M), para programar o microcontrolador Arduino MEGA2560 na linguagem de programação C. A programação do código G foi retirada do GRBL para dar lugar à programação do padrão STEP-NC baseado na norma (ISO 14649-1), permitindo assim que o arquivo no formato p21 (do padrão STEP) seja interpretado e enviado aos eixos de coordenadas da fresadora CNC. Uma conexão direta foi estabelecida entre o *software* STEP-NC Controller e a placa de 8 relés, para permitir que a fresadora CNC trabalhe de forma seletiva (código G e M ou STEP-NC). Além disso, uma operação remota da máquina via Internet foi implementada, possibilitando assim o envio do arquivo p21 via *web*, além do processo de usinagem poder ser acompanhado via *webcam* local.

Palavras-chave: STEP-NC, código G, Java, fresadora CNC, padronização.

ABSTRACT

The current standard programming of machine centers is still the (ISO 6983-1) standard, known as G-code, which was developed in the 1950s. This language contains low-level information performing the movement of the tool in the machining process, making it difficult for the operator to take decisions in machining. For this reason, the (ISO 6983-1) standard currently limits significantly the flexibility and interoperability part manufacturing. Given this scenario, in the 1990s a new data model for manufacturing processes was proposed to extend the STEP standard (ISO 10303-1), called STEP-NC (ISO 14649-1 standard), which was developed as a result of numerous projects by companies and universities. In this paper we describe the development and validation of software and hardware that allows the user to command a CNC gantry milling machine so that it will machine parts using the STEP-NC standard (ISO 14649-11). STEP-NC is based on the concept of workingsteps, machining *features* and machining operations, among other entities described by the standard (ISO 14649-1). The STEP-NC Controller software was developed using the Java programming language, which enabled the creation of a control interface for the CNC milling machining process. Programming compliant with the RS-274 standard (G and M codes) generated by the open source software GRBL (software and control movements on machines) was used to program the Arduino MEGA2560 microcontroller in the C programming language. G-code programming was removed from GRBL in order to make room for use of the STEP-NC standard (ISO 14649-1), thus allowing the physical file p21 (a part of the STEP standard) to be interpreted and sent to the axes of the CNC milling machine. A direct connection is established between the STEP-NC Controller and a board with 8 relays to allow the CNC milling machine to work selectively (either with G and M code or STEP-NC). Remote operation of the machine via the Internet was also implemented, allowing the sending of a p21 file via web, and the machining process can be monitored via webcam.

Keywords: STEP-NC, G-Code, Java, CNC milling machine, standardization.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 - Parâmetros do processo de fresamento frontal..... | 28 |
| Figura 2 - Exemplos de <i>features</i> | 30 |
| Figura 3 - Troca de dados entre sistemas CAD/CAM..... | 34 |
| Figura 4 - Linha de programa típica usando a norma ISO 6983 aplicando como referência o interpretador RS274NGC..... | 38 |
| Figura 5 - Estrutura de alto nível do STEP..... | 42 |
| Figura 6 - Exemplo de arquivo físico no formato STEP part 21..... | 43 |
| Figura 7 - EXPRESS-G representação do esquema geral da ISO 14649..... | 46 |
| Figura 8 - Três tipos de STEP-CNC..... | 47 |
| Figura 9 - Código G em comparação ao código STEP-NC..... | 50 |
| Figura 10 - Descrição geral do modelo de dados e o fluxo de dados..... | 51 |
| Figura 11 - Exemplo simples de uma parte do programa para teste de fresamento..... | 52 |
| Figura 12 - Parte do programa STEP-NC de um fresamento..... | 53 |
| Figura 13 - Fresadora Router CNC..... | 59 |
| Figura 14 - Fluxograma do processo de fabricação atual na fresadora CNC... | 60 |
| Figura 15 - CNC USB Controller..... | 61 |
| Figura 16 - CNC USB Controller MK1..... | 62 |
| Figura 17 - Caminho percorrido do pulso de 5V para o eixo-árvore (pulso e direção), para os <i>servo drivers</i> dos eixos X e Y, e <i>driver</i> do eixo Z da fresadora CNC..... | 63 |
| Figura 18 - Placa seletora Relé de 8 canais..... | 64 |
| Figura 19 - Arduino Mega2560..... | 66 |
| Figura 20 - Fluxograma de separação dos sinais (pulso, direção, GND e do eixo-árvore) na placa 8 relés..... | 68 |
| Figura 21 - Interface do software STEP-NC Controller..... | 73 |
| Figura 22 - <i>Serial connection</i> | 74 |
| Figura 23 - <i>File</i> | 74 |
| Figura 24 - <i>Network</i> | 75 |
| Figura 25 - <i>Machine Status</i> | 75 |
| Figura 26 - <i>Machine Settings</i> | 76 |
| Figura 27 - <i>Control</i> | 77 |
| Figura 28 - <i>Console</i> | 77 |
| Figura 29 - Comandos implementados para interpretação do padrão STEP-NC no arquivo <i>step_nc.c</i> | 79 |
| Figura 30 - Definição de parâmetros na biblioteca <i>step_nc.h</i> | 80 |
| Figura 31 - Programação da pilha e da fila no arquivo <i>step_nc_tools.c</i> | 81 |
| Figura 32 - Programação da pilha e da fila no arquivo (<i>step_nc_tools.h</i>)..... | 82 |
| Figura 33 - Programação realizada para habilitar 7 dos 8 relés da placa seletora de sinais..... | 83 |
| Figura 34 - Fragmento da sequência de programação do padrão STEP-NC para o fresamento circular..... | 86 |

| | |
|--|-----|
| Figura 35 - STEP-NC <i>server</i> . | 87 |
| Figura 36 - Fragmento da programação de conexão via rede do STEP-NC <i>server</i> . | 88 |
| Figura 37 - Mensagem na tela do servidor no arquivo “.bat”. | 89 |
| Figura 38 - Conexão manual com o STEP-NC <i>server</i> . | 89 |
| Figura 39 - Programa DUC para a conexão entre o software STEP-NC Controller e o servidor STEP-NC <i>server</i> . | 90 |
| Figura 40 - Imagem feita pela webcam da fresadora CNC. | 91 |
| Figura 41 - Fluxograma geral do padrão STEP-NC implementado na fresadora CNC. | 92 |
| Figura 42 - Fluxograma geral dos padrões implementados. | 93 |
| Figura 43 - Usinagem em espuma de PVC do padrão STEP-NC no processo de ranhura retangular. | 96 |
| Figura 44 - Usinagem em espuma de PVC do padrão STEP-NC no processo de furação. | 97 |
| Figura 45 - <i>Features</i> , CAPP e <i>Workplan</i> do processo de fresamento circular. | 98 |
| Figura 46 - MK1 Motor Connector. | 113 |
| Figura 47 - MK1 Connector. | 113 |
| Figura 48 - MK1 JOG Connector. | 114 |
| Figura 49 - MK1 LIMIT Connector. | 115 |
| Figura 50 - Configuração de <i>upload</i> do arquivo “.hex” no Arduino Mega2560. | 117 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 - Bibliotecas e arquivos de programação do programa GRBL. | 70 |
| Tabela 2 – <i>Features</i> funcionais do STEP-NC Modeler funcionais na fresadora CNC..... | 95 |

LISTA DE ABREVIATURAS E SIGLAS

- AP - Application Protocol (Protocolo de Aplicação)
- AVRDUDE - AVR Downloader Uploader
- CAD - Computer-Aided Design (Projeto Auxiliado por Computador)
- CAM - Computer Aided Manufacturing (Manufatura Auxiliada por Computador)
- CAE - Computer-Aided Engineering (Engenharia Auxiliada por Computador)
- CAPP - Computer-Aided Process Planning (Planejamento do Processo Auxiliado por Computador)
- CNC - Computer Numerical Control (Controle Numérico Computadorizado)
- FB - Function Block (Bloco de Função)
- FMC - Flexible Manufacturing Cells (Célula Flexível de Manufatura)
- FMS - Flexible Manufacturing System (Sistema Flexível de Manufatura)
- GND - Ground (Aterramento)
- GRBL - Software for controlling the motion of machines (Software de controle e movimentos em máquinas)
- IEC - International Electrotechnical Commission (Comissão Internacional de Eletrotécnica)
- IGES - Initial Graphics Exchange Specification
- ISO - International Organization for Standardization
- NC - Numerical Control (Comando Numérico)
- NIST - National Institute of Standards and Technology - Instituto Nacional de Padrões e Tecnologia
- STEP - Standard for the Exchange of Product Model Data (ISO 10303)
- STEP-NC - Standard for the Exchange of Product model data for Numerical Control (ISO 14649)
- UFSC - Universidade Federal de Santa Catarina
- XML - eXtensible Markup Language

SUMÁRIO

| | |
|---|-----|
| 1. INTRODUÇÃO | 23 |
| 2. OBJETIVOS DO TRABALHO | 25 |
| 2.1. Objetivo Geral | 25 |
| 2.2. Objetivos Específicos | 25 |
| 3. REVISÃO BIBLIOGRÁFICA | 27 |
| 3.1. Operações de Usinagem | 27 |
| 3.2. Sistemas Assistidos por Computador | 28 |
| 3.2.1. Computer-Aided Design (CAD)..... | 28 |
| 3.2.2. Projeto por features | 29 |
| 3.2.3. <i>Computer-Aided Manufacturig</i> (CAM)..... | 31 |
| 3.2.4. <i>Computer-Aided Process Planning</i> (CAPP)..... | 31 |
| 3.3. Integração de sistemas CAD/CAM | 32 |
| 3.4. Integração de sistemas CAM/CNC | 34 |
| 3.5. O Código G/M (ISO 6983) | 35 |
| 3.5.1. Programação de máquinas NC com o ISO6983..... | 36 |
| 3.5.2. O interpretador NIST RS274NGC..... | 37 |
| 3.5.3. Problemas com Código G..... | 38 |
| 3.6. O STEP (ISO 10303-1) | 40 |
| 3.7. O STEP-NC (ISO 14649)..... | 43 |
| 3.8. Integração CAD/CAPP/CAM/ CNC com STEP-NC | 50 |
| 3.9. Programação para operação de fresamento no padrão STEP-NC 52 | |
| 3.10. Projetos relacionados à norma STEP-NC (ISO 14649)..... | 55 |
| 4. MATERIAIS E MÉTODOS | 59 |
| 4.1. Conversão da fresadora CNC atualmente com código G para o padrão STEP-NC..... | 63 |
| 4.2. Configuração do hardware para que o CNC da máquina trabalhe de forma seletiva (CÓDIGO G ou STEP-NC) | 65 |
| 4.3. Interpretador de arquivo STEP-NC para geração de trajetória de ferramentas..... | 69 |
| 4.3.1. Análise do código G no programa GRBL | 70 |
| 4.3.2. Software desenvolvido para o controle da fresadora CNC para a envio do código STEP-NC | 71 |
| 4.3.3. Programação da máquina aderente ao padrão STEP-NC | 78 |
| 4.4. Operação remota da máquina via Internet..... | 87 |
| 5. VALIDAÇÃO EXPERIMENTAL POR USINAGEM..... | 95 |
| 6. CONCLUSÃO | 101 |
| 6.1. Considerações finais..... | 101 |
| 6.2. Sugestões para Trabalhos Futuros..... | 103 |
| 7. REFERÊNCIAS | 105 |

| | |
|---|-----|
| APÊNDICE A - CONFIGURAÇÃO DE PINOS DA PLACA CNC USB CONTROLLER MK1 | 113 |
| APÊNDICE B - DOWNLOAD E UPLOAD DO ARQUIVO “.HEX” NO ARDUINO MEGA2560 | 117 |
| APÊNDICE C - ARQUIVO STEP-NC CONSTRUÍDO PARA O TESTE DE FRESAMENTO RANHURA..... | 119 |
| APÊNDICE D - CÓDIGO G CONSTRUÍDO PARA O TESTE DE FRESAMENTO RANHURA..... | 123 |
| APÊNDICE E - ARQUIVO STEP-NC CONSTRUÍDO PARA TESTE DE FURAÇÃO E FRESAMENTO CIRCULAR..... | 125 |
| APÊNDICE F - CÓDIGO G CONSTRUÍDO PARA O TESTE DE FRESAMENTO RANHURA..... | 133 |

1. INTRODUÇÃO

Este trabalho de mestrado visa tornar uma fresadora CNC tipo pórtico (*router*) aderente ao padrão STEP-NC - norma (ISO 14649-1), utilizando um microcontrolador Arduino e uma placa de relés de 8 canais. Um arquivo *open source*, chamado GRBL, foi utilizado para reprogramar o código e adequar o padrão STEP-NC ao microcontrolador. O arquivo GRBL foi programado em linguagem C por um grupo de colaboradores para controlar fresadoras CNC de três eixos, utilizado o código G - norma (ISO 6983-1) / RS174D - para realizar operações de usinagem em fresadoras CNC.

Atualmente, as máquinas de controle numérico computadorizado (CNC) têm utilizado há mais de 50 anos o mesmo padrão de programação para a fabricação de peças, que corresponde à norma (ISO 6983-1) conhecido como código G (KEMMERER, 2009).

Essa linguagem se baseia na posição da extremidade da ferramenta, enviando instruções à máquina referentes aos movimentos que ela deve efetuar (ROSSO JR e NEWMAN, 2003). Estas informações correspondem a padrões de baixo nível, os quais têm sido usados como interface entre CAM e CNC. No entanto, a norma (ISO 6983-1) foca na programação do percurso da extremidade da ferramenta em relação aos eixos da máquina, em vez do processo de usinagem em relação à peça (ROSSO JR, 2005), (ALVARES, 2003).

Além disso, é inviável para o sistema de controle CNC executar um controle autônomo e inteligente que lide com possíveis necessidades de mudanças como, por exemplo, a alteração de uma superfície da peça, fazendo com que muitas informações referentes às estratégias de usinagem sejam perdidas, tornando difícil e/ou impossível o compartilhamento de tais informações (HARBS, 2012).

Diante destas limitações, foi concebido em âmbito internacional um novo modelo de dados entre sistemas CAD/CAM/CNC, conhecido como STEP-NC - norma (ISO 14649-1) - com o intuito de substituir a norma (ISO 6983-1) / RS174D, incluindo uma extensão da norma para os processos de fabricação (fresamento, torneamento, furação e eletro-erosão). Ela define um modelo de dados orientado a objetos para CNC, com uma estrutura detalhada de interface de dados que incorpora a programação baseada em *features* (ALVARES, 2003).

O padrão STEP-NC possui uma linguagem de alto nível que contém informações amplas sobre o processo de usinagem, permitindo não só a interoperabilidade dos sistemas (CAD/CAM/CNC) como

também o envio do arquivo (padrão STEP-NC) contendo as trajetórias de movimento da máquina via rede (*internet*) para o controlador da máquina, tornando a fabricação mais flexível e rápida quando comparado com o código G (BENAVENTE, 2011).

STEP-NC contribui para que as empresas acelerem suas tomadas de decisões referentes ao processo de fabricação devido à fusão dos sistemas CAD/CAM/CNC em um único *software*, além de permitir o compartilhamento de informações do processo como um todo, tanto internamente como via rede (*internet*) (SUH *et al.*, 2008).

2. OBJETIVOS DO TRABALHO

2.1. OBJETIVO GERAL

Conceber uma conversão da fresadora CNC, atualmente com código G, para trabalhar também no padrão STEP-NC, configurando um *hardware* para que a máquina trabalhe de forma seletiva (código G ou STEP-NC), realizando assim uma programação do padrão STEP-NC no código aberto GRBL, compilando-o no microcontrolador Arduino Mega2560 de forma a torná-lo aderente à norma (ISO 14649-11). Permite-se assim carregar o arquivo físico p21 (programa de peça em STEP-NC) de forma nativa, validando-o experimentalmente por usinagem.

2.2. OBJETIVOS ESPECÍFICOS

- Converter uma fresadora CNC tipo pórtico (*router*) atualmente com código G para operar também no padrão STEP-NC, para fins de comparação de qualidade de usinagem entre os dois métodos.
- Configurar o *hardware* para que o CNC da máquina opere de forma seletiva nos códigos G ou STEP-NC.
- Desenvolver um interpretador de arquivo STEP-NC para geração de trajetória de ferramentas.
- Implementar a operação remota da máquina via Internet.
- Validar o experimento por usinagem.

3. REVISÃO BIBLIOGRÁFICA

3.1. OPERAÇÕES DE USINAGEM

Dentre as operações de usinagem, o fresamento é o processo de usinagem na qual o material é removido por meio do movimento de vários gumes que cortam o material simetricamente em torno de um eixo. Cada gume remove uma pequena parcela de material da peça por meio da rotação da fresa em torno do eixo e pelo movimento de avanço, geralmente imposto pela peça. Neste caso, a ferramenta gira e a peça ou a ferramenta se deslocam em uma trajetória qualquer na máquina (SOUZA JÚNIOR, 2001).

Por proporcionar alta versatilidade no processo de usinagem, devido ao movimento da ferramenta de corte e da peça, isso possibilita a obtenção de cortes em várias direções. As condições de corte que se utilizam no fresamento são os principais responsáveis para se obter um bom acabamento superficial, aumentar a vida da ferramenta, evitar erros de forma e preservar a máquina, que influenciarão diretamente na qualidade da peça (SOUZA JÚNIOR, 2001).

Alguns parâmetros do processo de fresamento são descritos abaixo (ver Figura 1).

- Velocidade de corte (v_c em m/min): é a velocidade tangencial com que o diâmetro externo da fresa percorre a peça.
- Velocidade de avanço da mesa (f em mm/min): é a distância percorrida da peça em relação à ferramenta a cada minuto.
- Avanço por rotação (f em mm/rev): é a distância percorrida pela peça (mesa de trabalho) para cada rotação da fresa.
- Avanço por dente (f_z em mm/rev): é a distância percorrida pela mesa de trabalho para cada rotação da mesa dividido pelo número de dentes da fresa.
- Profundidade de corte (a_p em mm): é a penetração da ferramenta na peça, medida numa direção perpendicular ao plano de trabalho, é a profundidade com que a ferramenta de corte cortará a peça.
- Penetração de trabalho (a_e em mm): é a penetração da ferramenta em relação à peça, medido no plano de trabalho e numa direção perpendicular à direção de avanço.

- Penetração de avanço (a_f em mm): é a penetração da ferramenta medida no plano de trabalho e na direção de avanço.

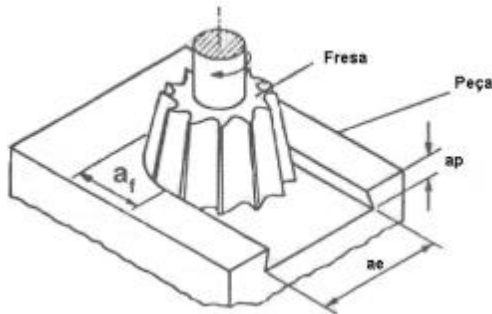


Figura 1 - Parâmetros do processo de fresamento frontal.

Fonte: (SOUZA JÚNIOR, 2001).

3.2. SISTEMAS ASSISTIDOS POR COMPUTADOR

Os sistemas assistidos por computador aproveitam a tecnologia da computação para facilitar muitos dos trabalhos realizados na área da engenharia, p.ex. desenho CAD, manufatura CNC, etc. A seguir é apresentada uma resenha básica de alguns dos sistemas assistidos por computador utilizados na área da manufatura e de fabricação.

3.2.1. Computer-Aided Design (CAD)

Segundo (BROGUEIRA, LOPES e KLEIN, 2004), no CAD (*Computer-Aided Design*) todos os elementos de *hardware* e *software* se fundem em uma única “ferramenta” para o desenho, podendo ser definido como um *software* que dá suporte ao projeto e ao processo de engenharia do produto. Nesta definição geralmente estão incluídas as ferramentas computacionais classificadas como Projeto Assistido por Computador (CAD) e a Engenharia Assistida por Computador (CAE), que contribuem para a modelagem de produtos, componentes e no detalhamento de desenhos.

Na indústria, os *software* CAD são amplamente utilizados para o projeto de produtos no auxílio à criação, modificação, análise ou

otimização de um projeto. O CAD é um recurso utilizado para aumentar a produtividade do projetista, melhorando a qualidade do desenho e a comunicação através de documentações e bancos de dados, aos quais os responsáveis pela produção têm acesso (BROGUEIRA, LOPES e KLEIN, 2004).

O *software* CAD utiliza gráficos que são baseados em vetores para representar os objetos que estão sendo desenhados, o que possibilita a visualização das curvas do desenho em duas dimensões do espaço (2D) ou curvas e superfícies e sólidos em três dimensões no espaço (3D), mostrando assim a real aparência do objeto que foi projetado (BIZELLO, 2001). O CAD é também aplicado em *designs*, principalmente para desenhos industriais, projetos automotivos, projetos de máquinas CNC, robótica, aeroespacial entre outros. É utilizado nas produções de filmes com animações gráficas e efeitos especiais. Devido à sua contribuição para o ambiente de manufatura, o CAD tem grande importância na computação gráfica (*hardware* e *software*), bem como na geometria diferencial discreta (NARAYAN et al., 2008).

Apesar dos computadores atuais e dos *software* CAD terem se tornado extremamente poderosos, o CAD é pouco usado por engenheiros que não compreendem plenamente os princípios fundamentais gráficos e estratégias de modelagem em 3D (RYNNE e GAUGHRAN, 2008). Entretanto, os sistemas CAD estão sempre em constantes mudanças, pois não possuem uma ferramenta totalmente acabada e com todas as funcionalidades que um projetista precisa para a fase de criação e idealização de um projeto. Atualmente exige-se que o CAD abranja e integre todo o desenvolvimento de um produto, representando-o no computador com uma série de opções, propriedades e atributos para que o projetista possa manipular e organizar a melhor forma de gerenciamento e recursos, os quais influenciam no produto final (MIAO, SHAH e SRIDHARAN, 2002).

3.2.2. Projeto por features

Em um sistema CAD as *features* são uma forma de representar informações presentes no desenho de uma peça, montagem ou de um processo de manufatura. As *features* são representadas mediante entidades geométricas simples, e para efetuar esta representação normalmente implanta-se um modelo de dados que permite relacionar tanto a geometria como a dados tecnológicos envolvidos. Esse modelo baseia-se no conceito de *features* básicas e modificadoras, onde dentre as

básicas tem-se os eixos e furos, enquanto as modificadoras contêm alterações nas *features* básicas, tais como: chanfro, arredondamento, concordância, rosca. As *features* básicas estão ligadas diretamente à peça, e cada *feature* básica possui uma lista de *features* modificadoras que podem ser associadas de acordo com a necessidade, tendo-se assim um modelo de produto com mais informações de alto nível do que somente a geometria explícita (MAZIERO, BONATTO e PEREZ, 2001). A Figura 2 mostra exemplos de *features*, como: furo, cavidade, chanfro e ranhura.

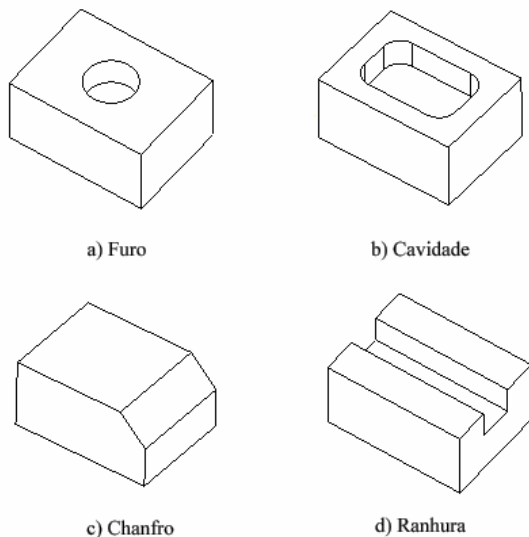


Figura 2 - Exemplos de *features*.

Fonte: Adaptado de Benavente (2007).

Na engenharia, as *features* possuem formas que podem ser descritas como parâmetros dimensionais, enumeração geométrica e entidades topológicas (BENAVENTE, 2011). Outro conceito de *features* consiste em características que dizem respeito às formas básicas e atributos tecnológicos que são associados a operações de manufatura, onde se pode avaliar o comportamento da peça em diversas situações, identificar ações que precisam ser feitas, e como tais ações que serão efetuadas (ROSSO JR, 2005).

3.2.3. Computer-Aided Manufacturig (CAM)

A Manufatura Assistida por Computador (CAM) é um sistema computacional que utiliza a tecnologia informática na fabricação e controle numérico (NC) para a automatização de um processo de manufatura e de controle numérico computadorizado (CNC), (GROOVER, 2007).

Sistemas CAM geram a programação NC no qual utilizam recursos computacionais no planejamento e controle de unidades na produção, e que trabalham tendo como base modelos matemáticos provenientes do sistema CAD, que fornece dados do projeto para o sistema CAM, que por sua vez geram um arquivo contendo a trajetória da ferramenta para o pós-processador CNC (*software* que gera o programa do comando específico da máquina) (SIVITTER, 2004).

O CAM sofreu grandes mudanças nos últimos 10 anos, mudanças estas ocasionadas, principalmente, pela evolução dos computadores, das ferramentas de usinagem e dos sistemas CAD. No princípio, os *softwares* CAD continham uma composição de linhas e superfícies, e o CAM era desenvolvido para usinagem 2D ou 3D, e ambos apresentavam um desempenho razoável nos dois processos. Uma das características mais importantes do CAM é a habilidade de prover caminhos de ferramentas sem que haja colisões, erros de planejamento da usinagem e futuras vibrações, proporcionando um acabamento ideal para superfícies irregulares (GAO, GINDY e ZHENG, 2004).

3.2.4. Computer-Aided Process Planning (CAPP)

A ponte que liga o projeto e a fabricação é o planejamento do processo. O Planejamento de Processo Assistido por Computador (CAPP) é um processo que determina instruções detalhadas da operação para transformar um projeto de engenharia em um produto final (CHANG, WANG e WYSK, 2005). O tamanho, a forma, as tolerâncias e o acabamento afetam os planejamentos de processo. A tarefa de planejamento do processo envolve muitas atividades, como a seleção de operações de usinagem, ferramentas de corte, parâmetros de corte e sequência de operações. Todas as atividades estão intimamente relacionadas e dependem umas das outras. No procedimento manual de planejamento do processo, geralmente ocorrem problemas, como falta de experiência do projetista, inconsistência dos planos de processos e a necessidade de automatizar a tarefa de planejamento de processo.

O CAPP contribuiu para simplificar e melhorar o planejamento de processo, incluindo a disponibilização de informações sobre o uso de produtos e recursos, deixando a produção mais eficaz. O objetivo do CAPP consiste em gerar um conjunto de instruções utilizadas para fabricar a peça, que pode ser aplicado juntamente com o CAM (CHANG, WANG e WYSK, 2005).

O CAPP é visto como um agente de comunicação entre os sistemas CAD e CAM. Para isso são necessários dois tipos de informações: os dados geométricos, que descrevem o desenho de uma peça, e os dados tecnológicos, que são utilizados para a fabricação da peça (HAN, HAN e YI, 2001).

Em função dos recursos que o CAPP proporciona, os pesquisadores puderam dar mais atenção ao processo de fabricação devido ao importante papel na integração CAD/CAM. O CAPP é capaz de fornecer uma base adequada para a integração de projetos e etapas posteriores, tais como a análise de engenharia, a usinagem e a inspeção (HAN, HAN e YI, 2001).

Baseando-se em recursos que interpretam o modelo do produto em termos de *features* de usinagem, gera-se instruções para produzir o produto. Por exemplo, o CAPP normalmente contém recursos para definir qual a melhor a estratégia para o processo de furação. O sistema se autocorrigue através das informações obtidas dos setores (*feedback*) como, por exemplo, a correção do tempo de usinagem e do tempo de vida das ferramentas utilizadas no processo, através das informações passadas ao CAPP pelo próprio CNC (TANAKA e KISHINAMI, 2006).

3.3. INTEGRAÇÃO DE SISTEMAS CAD/CAM

Os sistemas CAD/CAM necessitam trocar ou compartilhar dados e informações armazenadas em formatos de arquivos ou bases de dados computacionais. Em uma troca de dados ideal, deve-se abranger a descrição completa do produto em seu banco de dados, e os principais tipos de dados ou informações que compõem essa descrição são: dados geométricos (forma), dados não geométricos, dados de projeto, dados de fabricação, e dados adicionais das atividades subsequentes do ciclo de vida do produto (DIAS e MATOS DA CUNHA, 2000).

Apesar das *features* desempenharem um papel importante na integração CAD/CAM, elas compartilham a mesma geometria do modelo de um produto em um formato neutro ou proprietário. No entanto, a informação referente ao plano de processo a partir de sistemas CAPP não

pode tornar-se um ponto de partida para que os sistemas CAM gerem caminhos da ferramenta, uma vez que o usuário teria que criar manualmente as operações de usinagem e definir a geometria, ferramenta de corte e vários parâmetros para cada operação a ser desempenhada. (ADAMCZYK e KOCIOLEK, 2001).

Os dados de projeto CAD não podem ser utilizados diretamente em sistemas CAM, por conta da variedade de recursos de ponto de vista e da representação característica para cada aplicação. Recurso é o meio de transmissão de informações no âmbito de CAD/CAM e CAPP que transporta uma quantidade significativa de informações, tanto geométricas como não-geométricas, e organicamente liga cada sistema de CAD, CAPP e CAM (MIAO, SHAH e SRIDHARAN, 2002).

No entanto, as definições de recursos dependem de diferentes pontos de vista da aplicação. As características de projeto referem-se à forma e à função, incluindo a intenção (GAO, GINDY e ZHENG, 2004).

Uma das alternativas para efetuar-se a troca de dados entre sistemas CAD/CAM era baseada em tradutores de arquivos mediante algoritmos para cada formato a ser utilizado, convertendo os dados geométricos para um formato aceitável para o outro aplicativo.

Com o avanço das pesquisas surgiram formatos de arquivos neutros como o IGES (*Initial Graphics Exchange Specification*) norma (ANSI 1978), com o objetivo de reduzir o número de algoritmos tradutores. Já a aplicação baseada em formatos neutros limita o número de algoritmos tradutores na troca de dados, onde um constrói e armazena o formato genérico, enquanto que o outro apenas lê o formato. Uma comparação entre essas duas formas de troca de dados é mostrada na Figura 3, que contém exemplos da tradução direta e do formato neutro (DIAS e MATOS DA CUNHA, 2000). Por mais simples que seja a implementação da tradução direta, o arquivo neutro é o método mais adequado, pois conduz a uma padronização mais eficiente quando comparado com a direta, além da quantidade de tradutores ser menor. Um exemplo disso é um programa CAD com uma nova versão exigir que todos os tradutores relacionados a ele sejam atualizados também, o que não ocorre com a abordagem baseada em arquivo neutro.

A tradução direta tende a ser de mais fácil implementação, mas possui um custo maior para manter-se. Assim, o uso do arquivo neutro é o método que conduz a uma possível padronização muito mais facilmente.

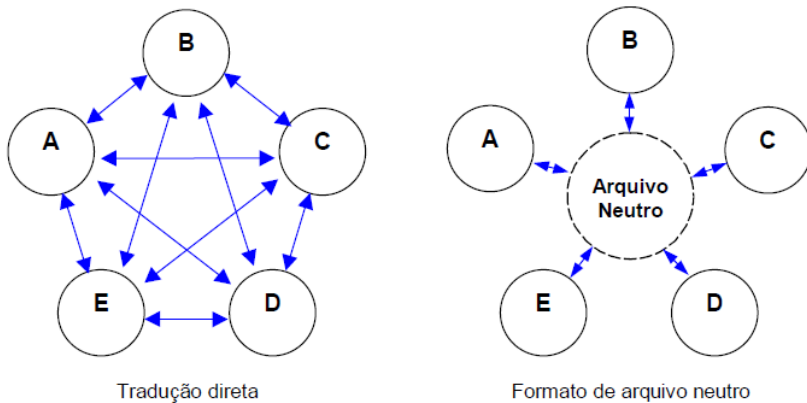


Figura 3 - Troca de dados entre sistemas CAD/CAM.

Fonte: (DIAS e MATOS DA CUNHA, 2000).

3.4. INTEGRAÇÃO DE SISTEMAS CAM/CNC

A flexibilidade existente nestes sistemas, em particular a flexibilidade das máquinas CNC, ainda é considerada limitada (LEONDES, 2002).

O controle do CNC atual é considerado inflexível, e isto ocorre devido ao uso da linguagem de programação ISO 6983 (código G), que se concentra na programação da trajetória da ferramenta com relação aos eixos da máquina, ao invés das tarefas serem executadas em relação à peça (XU et al., 2006).

Um *software* CAM apresenta uma série de funções que servem de auxílio ao programador para definir o caminho da ferramenta, assim como informações das ferramentas de corte que serão utilizadas no processo de usinagem da peça. O código G é o padrão de dados utilizado para representar as trajetórias da ferramenta para usinar uma peça em uma máquina CNC (ROSSO JR, 2005).

Entretanto, existe um problema de interoperabilidade neste processo, que faz com que durante essa integração CAM/CNC não haja um retorno das informações que são enviadas para o CNC. Dentre essas informações tem-se a estratégia de usinagem adotada pelo operador, tolerâncias geométricas da peça a ser usinada diante do espaço físico de usinagem na fresadora CNC, parâmetros de corte que são responsáveis

pelo processo de usinagem, assim como também os comandos de operação das máquinas (ROSSO JR, 2005).

Features de manufatura são derivadas das *features* geométricas onde geram OPERATIONS, WORKINGSTEPS e WORKPLANS para o processo de usinagem no padrão STEP-NC. Os arquivos STEP-NC são então transferidos para um tradutor, que converte-os para um formato proprietário do controlador baseado em *features* (SUH *et al.*, 2008).

A utilização do tradutor no controlador CNC pode ser considerado como um passo intermediário para realização do controlador STEP-NC inteligente. A vantagem desta aproximação é que qualquer mudança no chão de fábrica é possível de ser retransmitida para o tradutor para o arquivo STEP-NC, e de lá para o sistema CAD. Neste sistema, como a tradução entre geometria CAD e *features* de manufatura é manipulada no sistema CAM, não é necessário um controlador com grande capacidade computacional (HARBS, 2012).

Tendo em vista estas limitações, nos anos 1990 passou a ser desenvolvido um novo padrão para troca de dados com base na norma (ISO 10303-1), posteriormente denominada de (ISO 14649-1) ou STEP-NC (ROSSO JR e NEWMAN, 2003).

Segundo (HARBS, 2012), este novo padrão fornece um modelo de dados orientado a objetos para a nova geração de dispositivos CNC, pois ele utiliza a geometria da peça baseando-se no conceito de *features*. Além de conter informações que incluem dados da geometria da peça, das ferramentas a serem utilizadas, das sequências de operações que serão realizadas e do plano de trabalho, a norma (ISO 14649-11) ainda possibilita a troca bidirecional de dados entre sistemas CAM/CNC, tornando o processo mais fácil e rápido de ser modificado caso necessário devido a interoperabilidade no qual permite se comunicar de forma rápida entre os sistemas CAD/CAM/CNC, uma vez que os três podem estar contidos em um único *software*.

3.5. O CÓDIGO G/M (ISO 6983)

O código G - norma (ISO 10303-1) - também conhecido como RS274D, é o nome comum utilizado para referir-se à linguagem de programação de controle numérico (CN). O padrão define um conjunto de códigos G/M que especificam uma sequência de movimentos da ferramenta de corte, bem como o sentido de rotação, a velocidade de deslocamento e as funções auxiliares. O código G é uma linguagem na qual o programador define parâmetros do que fazer e como fazer para que o CNC possa realizar a usinagem da peça. Porém, a reduzida flexibilidade

desta norma não garante o intercâmbio de programas entre máquinas (XU e HE, 2004).

3.5.1. Programação de máquinas NC com o ISO6983

A programação de máquinas NC usando o padrão ISO 6983 resulta em programas extensos e difíceis de manipular. “Mudanças e correções de última hora de programas complexos são difíceis de gerenciar no chão de fábrica e o controle da execução do programa na máquina é muito limitado (STARK, KHAI e MAEDER , 2002).

A fim de simplificar as atividades de programação e para fazer com que os programas NC sejam mais compactos, a norma ISO 6983 permite um conjunto especial de código G chamado de ciclos fixos, que são funções pré-programadas. Como por exemplo, ciclos para furação profunda com quebra de cavaco (G83) (também conhecida como estratégia “pica-pau”), entrando e saindo para facilitar a quebra e remoção de cavacos em operações de furação profunda (ROSSO JR, 2005).

Assim como existe o ciclo fixo G83 para furação, também existem outros ciclos específicos implementados por cada fabricante de máquinas CNC.

Outra característica da norma ISO6983 é que ela permite a implementação de subprogramas e macros. Subprogramas são conjuntos de linhas de código que se usam várias vezes em um ou mais programas principais. A implementação destes subprogramas é específico de cada fabricante e tem diferenças entre controladores (ROSSO JR, 2005).

Estas variações de programação dos ciclos fixos e dos subprogramas que, segundo (STARK, KHAI e MAEDER , 2002), correspondem a mais de 5000 “dialetos”, resulta em problemas de interoperabilidade entre máquinas CNC. Portanto, é necessária uma abordagem padronizada que permita a interoperabilidade na manufatura CNC, e é por esta razão que os padrões relacionados à fabricação evoluíram (ROSSO JR e NEWMAN, 2003).

Existe um padrão chamado RS274 que é baseado no padrão ISO6983. Esse padrão foi uma revisão da *part 1* da norma ISO6983 aprovada em 1980. Assim como o padrão RS274 baseia-se na norma ISO6983, há outras modificações ou revisões ou modificações que tomam nomes diferentes, p.ex. BLC, RS274NGC, DIN, entre outros.

Como já foi mencionado, o padrão RS274 (a versão mais atual é chamada RS274D) possui algumas modificações com relação com o ISO6983, no entanto neste texto é usado o termo código G ou G/M como

referência tanto do padrão ISO6983 quanto o padrão RS274 e o RS274NGC.

3.5.2.O interpretador NIST RS274NGC

Este intérprete lê o código de controle numérico NC (ISO 6983 ou Código G/M) e produz as chamadas para um conjunto de funções canônicas para realizar a usinagem. O intérprete é um *software* escrito em C++. A saída do intérprete pode ser usada para movimentar centros de usinagem de três a seis eixos. A entrada para o intérprete RS274 é código no dialeto definido pelo controlador de Próxima Geração (NGC ou RS274NGC), que é um código G/M com algumas modificações (KRAMER, MESSINA e PROCTOR, 2010).

O intérprete pode ser compilado como um programa de computador autônomo ou pode ser integrado com o controlador da máquina. Para a entrada do intérprete é possível inserir um arquivo físico com código G/M ou um usuário digitando inteiramente o código G/M. Os comandos de saída podem ser impressos para uso futuro ou podem ser executados diretamente em um centro de usinagem. A Figura 4 mostra um fragmento de uma linha de programação NC.

Existe ainda um relatório que inclui uma descrição completa da linguagem de entrada RS274/NGC e as funções canônicas de usinagem chamados pelo intérprete (KRAMER, MESSINA e PROCTOR, 2010).

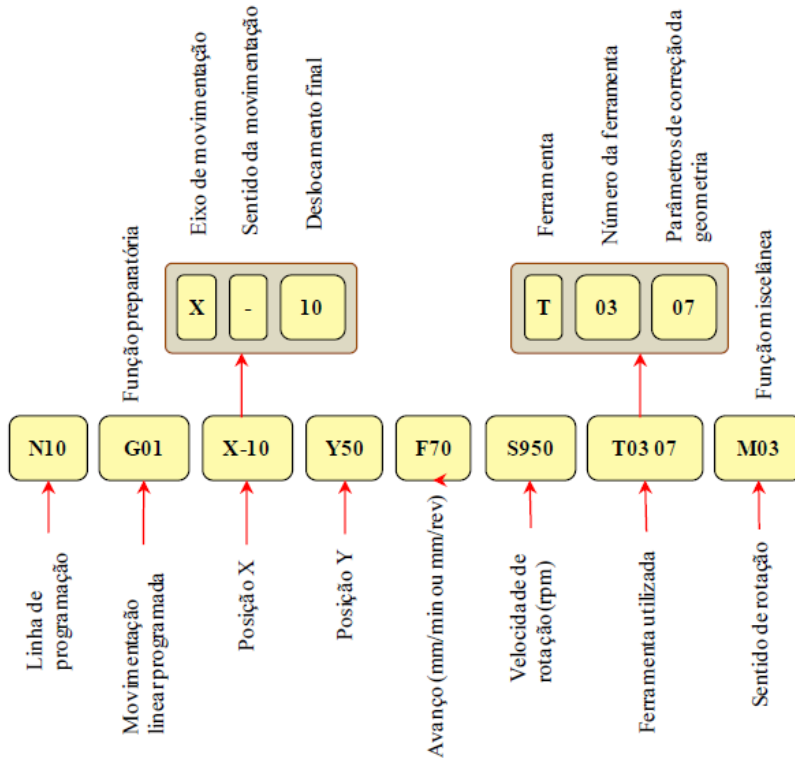


Figura 4 - Linha de programa típica usando a norma ISO 6983 aplicando como referência o interpretador RS274NGC.

Fonte: (BENAVENTE, 2011).

3.5.3. Problemas com Código G

A usinagem de alta velocidade, de alta precisão, de multi-eixos e a usinagem complexa aumentaram tanto a produtividade quanto a qualidade de fabricação. A linguagem de programação do código G foca o caminho do centro da fresa (*center location* - CL) em relação aos eixos da máquina, ao invés das funções de usinagem em relação à peça (SUH *et al.*, 2008).

Além disso, o padrão define a sintaxe de instruções do programa, mas na maioria dos casos a semântica é incompleta, fazendo com que a linguagem seja complementada com extensões que não são incluídas na

norma (ISO 6983-1), dificultando a leitura e execução do programa nas máquinas CNC devido a cada máquina possuir um pós-processador dedicado e próprio. O código G suporta apenas uma forma de fluxo de informações do projeto à fabricação, do projeto CAD para a manufatura CNC no qual não permite mudanças durante o processo de fabricação o chão de fábrica, impossibilitando a inclusão de novos parâmetros. (SUH *et al.*, 2008).

Assim, importantes informações do processo de usinagem dificilmente serão preservadas, sendo muito difícil mudar um programa, pois os dados do CAD não são utilizados diretamente na máquina. Assim, os dados devem ser processados por um pós-processador, mediante o qual se obtém apenas um conjunto de baixo nível de instruções e dados incompletos, dificultando a verificação e a simulação da usinagem. Os problemas normalmente encontrados na norma ISO 6983 são resumidos abaixo (SUH *et al.*, 2008).

- Perda de informações: um programa contendo os códigos G/M é definido por um código alfanumérico que contém caracteres como os seguintes: G, T, M, F, S, indicando o movimento da máquina e o eixo para o controlador. O programa disponibiliza ao CNC apenas informações limitadas, excluindo informações valiosas como a geometria da peça e o plano de processo usado para gerar o código NC.
- Rastreabilidade difícil: Como um programa de códigos G/M é composto por um conjunto codificado de números para movimentos de eixo, não é fácil para os operadores de máquinas compreenderem o fluxo operacional, devido à falta de informações, condição de usinagem e das especificações da ferramenta. A ausência destas informações gera perda de dados entre os sistemas CAD e CAM e, em particular, torna-se mais difícil não só encontrar qual parte do programa possa causar problemas como também modificá-las para o processo de usinagem
- Falta de interoperabilidade: Os códigos G e M dependem muitas vezes da compatibilidade na máquina do fabricante ou do controlador. Por exemplo, os comandos G70 e G80 significam ciclos de usinagem, enquanto G98 corresponde ao avanço por minuto na série 0 da FANUC. Por outro lado, G60 e G80 significam códigos de ciclo, G94 instrui avanço

por minuto na série 8055T da FAGOR. O programa origina-se de um controlador de um determinado código e normalmente não pode ser usado em outro controlador. Por esta razão, é necessário aplicar um pós-processamento para que o programa se adapte a uma configuração específica da máquina CNC. Este pós-processamento é uma das principais interferências do fluxo de dados integrado na cadeia de CAD-CAM-CNC.

- Não compatibilidade com os sistemas de níveis mais elevados: Em sistemas de fabricação de alto nível, como o CAD/CAPP/CAE/CAM/PDM/MRP, a troca de informações compatíveis está sendo aumentada gradualmente com a introdução da tecnologia STEP (norma ISO 10303). Em contraste, o ambiente rico em informações quase pereceu no CNC no nível de chão de fábrica. Além disso, há poucas informações de *feedback* do CNC, o que torna a situação no chão de fábrica obscura para os sistemas nos níveis mais altos. Inevitavelmente, o CNC no chão de fábrica continua a ser uma ilha isolada na cadeia de CAD-CAM-CNC.

3.6. O STEP (ISO 10303-1)

Em julho de 1984 foi realizada a primeira reunião do subcomitê ISO TC 184/SC4 no Instituto Nacional de Padrões e Tecnologia (NIST). Participaram representantes de países tais como Canadá, França, Alemanha, Suíça, Inglaterra, e Estados Unidos. O motivo desta reunião foi criar um padrão internacional que permitisse capturar a informação do modelo computadorizado do produto de uma forma neutra sem perda da integridade, através do ciclo de vida do produto.

STEP é o acrônimo de “*Standard for the Exchange of Product Model Data*”, e descreve a coleção de documentos padrão publicados sob a norma ISO 10303.

A referência inicial do STEP foi proporcionar a coleção de todos os dados do modelo do produto. A ideia de modelo de produto foi derivada da utilização dos sistemas de projeto assistido por computador (CAD). A segunda referência foi uma troca filosófica de “dados do produto”, implicando que qualquer quantidade de dados do produto poderia ser intercambiada (NIST, 1999).

STEP fornece uma representação da informação do produto junto com os mecanismos e definições necessários para permitir que os dados do produto sejam intercambiados. O intercâmbio é entre ambientes e sistemas computacionais diferentes associados ao ciclo de vida completo de um produto, incluindo projeto, manufatura, utilização, manutenção e descarte. A informação do produto gerada durante esse processo pode ser usada para diferentes propósitos. Isto pode envolver diversos sistemas de computadores, inclusive alguns podem estar localizados em organizações diferentes. Para suportar tais usos, as organizações devem ser capazes de representar a informação do produto de uma forma interpretável, e é necessário conservar tal informação completa e consistente quando intercambiada entre diferentes sistemas (SCRA, 2006).

Todos os esforços para padronização da troca de dados de produtos têm convergido para o padrão STEP, norma (ISO 10303-1) para a definição de um padrão de modelo computadorizado para troca de dados e informações do produto, em todo o seu ciclo de vida. De acordo com (EASTMAN, 1994), os objetivos do STEP são:

- Armazenar todos os dados usados para a descrição do produto;
- Armazenar todo o ciclo de vida do produto;
- Armazenar formas diversas de utilização dos dados do produto;
- Compartilhar dados de forma precisa e não incompleta;
- Compartilhar dados entre aplicações específicas;
- Compartilhar dados sem restringir a forma como uma indústria possa utilizar esses dados. Assim, a indústria pode adotar processos otimizados e mais ajustados às necessidades específicas.

A seguir são apresentadas as melhorias proporcionadas pelo padrão STEP em comparação com as deficiências do padrão IGES (BROWNE e MCMAHON, 1998).

- O STEP (ISO 10303-11) usa um modelo de dados, o qual é descrito utilizando uma linguagem formal de especificação que contém informações chamado EXPRESS, que foi desenvolvido especialmente para o STEP. A linguagem EXPRESS é legível tanto pelo usuário como processável pelo computador. No STEP os dados são descritos pela linguagem EXPRESS, a qual mapeia o modelo para um arquivo físico. Neste caso, o arquivo físico

pode definir como cada entidade em particular deve ser representada, que é o que acontece com o padrão IGES.

- O STEP possui uma arquitetura de camadas em três níveis, essas camadas habilitam a visão múltipla de aplicação e implementação a serem definidas.

O padrão STEP está organizado em uma série de partes, cada uma publicada separadamente. Estas partes incluem uma das seguintes séries: métodos de descrição, recursos integrados, protocolos de aplicação, conjuntos de testes abstratos, métodos de implementação, e testes de conformidade (SCRA, 2006 e LOFFREDO, 2000).

LOFFREDO (2000) divide a estrutura em duas partes: Infraestrutura e Modelos de informação, como pode-se ver na Figura 5.

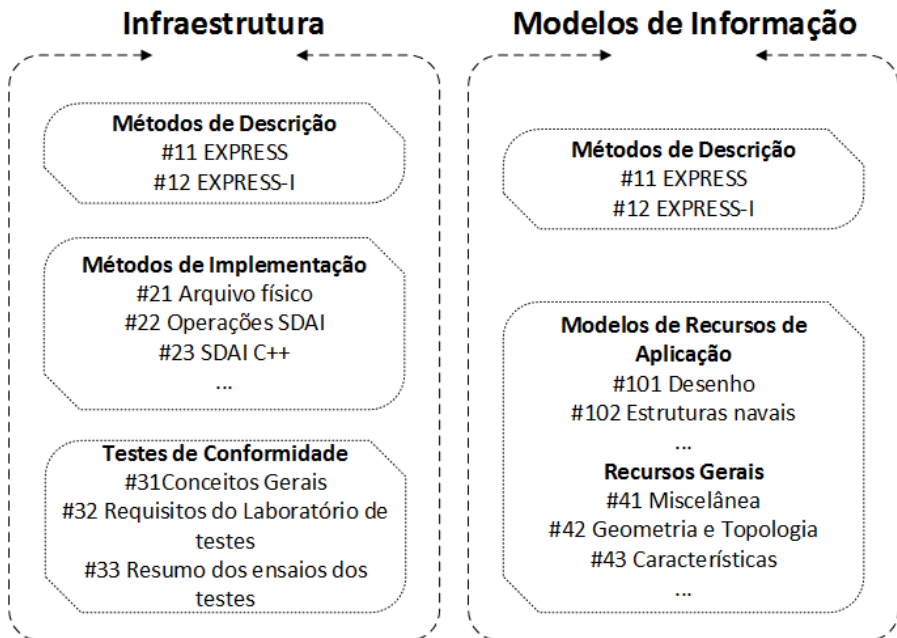


Figura 5 - Estrutura de alto nível do STEP.

Fonte: Adaptado de (LOFFREDO, 2000)

A Figura 6 mostra um exemplo de arquivo físico no formato STEP (ISO 10303-21).

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(
/* description */ ('A minimal AP214 example with a single part'),
/* implementation_level */ ('2;1');
FILE_NAME(
/* name */ ('demo',
/* time_stamp */ ('2003-12-27T11:57:53',
/* author */ ('Lothar Klein'),
/* organization */ ('LKSoft'),
/* preprocessor_version */ ('',
/* originating_system */ ('IDA-STEP',
/* authorization */ (''));
FILE_SCHEMA (('AUTOMOTIVE_DESIGN { 1 0 10303 214 2 1 1}'));
ENDSEC;
DATA;
#10=ORGANIZATION('00001', 'LKSoft', 'company');
#11=PRODUCT_DEFINITION_CONTEXT('part definition', #12, 'manufacturing');
#12=APPLICATION_CONTEXT('mechanical design');
#13=APPLICATION_PROTOCOL_DEFINITION('', 'automotive_design', 2003, #12);
#14=PRODUCT_DEFINITION('0', $, #15, #11);
#15=PRODUCT_DEFINITION_FORMATION('1', $, #16);
#16=PRODUCT('A0001', 'Test Part 1', '', (#18));
#17=PRODUCT_RELATED_PRODUCT_CATEGORY('part', $, (#16));
#18=PRODUCT_CONTEXT('', #12, '');
#19=APPLIED_ORGANIZATION_ASSIGNMENT(#10, #20, (#16));
#20=ORGANIZATION_ROLE('id owner');
ENDSEC;
END-ISO-10303-21;|
```

Figura 6 - Exemplo de arquivo físico no formato STEP part 21.

Fonte: (ISO 10303-21)

3.7. O STEP-NC (ISO 14649)

O CNC vem se tornando cada vez mais complexo, exigindo um controle descentralizado com a sua estrutura interna flexível e personalizável (MINHAT *et al.*, 2009). O surgimento do STEP-NC não pode ser ignorado quando se trata de um modelo para o compartilhamento de informações entre sistemas CAD/CAM. O STEP-NC (ISO 14649-1) é uma extensão do STEP (ISO 10303-1) a qual fornece não só uma descrição completa da peça, mas do processo de fabricação, contendo os dados de projeto CAD e informações sobre o processo de fabricação, suas

características de corte e seus requisitos de ferramenta, além de representar um modelo de transferência de dados entre sistemas CAD/CAM/CNC (YUSOF, TAN e KASSIM, 2009).

O STEP-NC define os dados que representam as etapas de trabalho, uma biblioteca de operações de usinagem específicas realizadas no CNC, para que qualquer controlador CNC seja capaz de calcular a trajetória da ferramenta com base em definições contidas em rotinas decompostas e integradas dentro do próprio controlador (LEWIS, 2002).

Ele foi desenvolvido para substituir o ISO 6983 (código G), permitindo a conexão de dados utilizados no projeto de CAD, e para determinar dinamicamente as exigências de usinagem para uma operação com os dados do processo de CAM (SUH *et al.*, 2002).

Este novo padrão proporciona uma metodologia abrangente para a representação de dados de fabricação, corrigindo os problemas encontrados na norma ISO 6983, mostrando cada passo da usinagem ao invés do movimento da ferramenta em relação à máquina utilizando o conceito orientado a objeto denominado *Workingstep*. Um *Workingstep* é uma entidade específica que pode ser executada em uma máquina-ferramenta CNC, onde os controladores CNC são responsáveis por traduzirem os *Workingsteps* em movimentação do eixo e operações da ferramenta (ROSSO JR e NEWMAN, 2003).

O padrão STEP-NC fornece uma entrada direta para as máquinas-ferramenta CNC, consistindo de informações sobre o produto, tais como geometria, características, etapas de usinagem e caminhos da ferramenta. Como mencionado anteriormente, atualmente as máquinas CNC são programadas usando o código G, que apenas descreve os movimentos exatos da ferramenta, sem qualquer informação da peça que está sendo processada. Ao contrário do código G, o STEP-NC informa a máquina o que fazer ao invés de como fazê-lo. Cada etapa de trabalho está relacionada a uma operação e uma *feature* de fresamento. Espera-se que controladores CNC venham a traduzir etapas de trabalho em movimentos dos eixos e da ferramenta para a fabricação da peça (TANAKA e KISHINAMI, 2006).

Com o uso de STEP-NC, muitos gargalos para a produtividade de fabricação serão eliminados, como pós-processadores complicados, programas NC (código G) antiquados, redundâncias de dados etc.. Hoje, as máquinas CNC não disponibilizam quaisquer informações sobre a peça e o processo. Com os controladores baseados em STEP-NC, pode ser passada a informação completa sobre a peça, o modelo 3D anotado com tolerâncias de projeto, recursos de fabricação, sequência de processo e requisitos de ferramentas de corte. O STEP-NC permitirá que máquinas-

ferramenta implementem estratégias alternativas para a fabricação da peça. Pós-processadores serão incorporados nos controladores inteligentes para que sejam transparentes para os usuários (MATTHIEU *et al.*, 2011).

O padrão STEP-NC (ISO 14649-1) é uma representação baseada em recursos estruturados de planos de processo para processos de fabricação, tais como operação de fresamento, eletro-erosão (EDM), etc. A (ISO 14649-11) foi desenvolvida para a operação de fresamento com base nas informações geométricas da norma (ISO 10303-1), tais como AP 203 e AP 224. O processo de fresamento no padrão STEP-NC baseou-se nas normas (ISO 14649-10), (ISO 14649-11) e (ISO 14649-111), o qual é composto por quatro partes: (A) descrição de tarefas, (B) descrição da tecnologia, (C) descrição da ferramenta, e (D) descrição geométrica, como mostrado na Figura 7. A descrição de tarefas descreve a sequência lógica de tarefas executáveis tais como (*machining_workingstep*, *NC_function*) e tipos de dados. Os detalhes de cada *workingstep* são abordados na descrição da tecnologia com referência à descrição da ferramenta e a descrição da geometria (SUH *et al.*, 2003).

Especificamente, os *workingsteps* incluem características de fabricação para 2.5D (*two5D_manufacturing_feature*) e para as operações de fresamento em 3D (*region*), e cada *workingstep* possui subclasses tais como: (*planar_face*, *pocket*, *step*, *slot*, *round_hole* e *general_outside_profile*) em conjunto com as condições de corte contidas no arquivo p21. É importante notar que o percurso da ferramenta no processo de fresamento no padrão STEP-NC (ISO 14649-11) é "opcional", ao contrário da atual programação NC em que a trajetória da ferramenta em termos de eixos da máquina é o principal conteúdo da informação (SUH *et al.*, 2003).

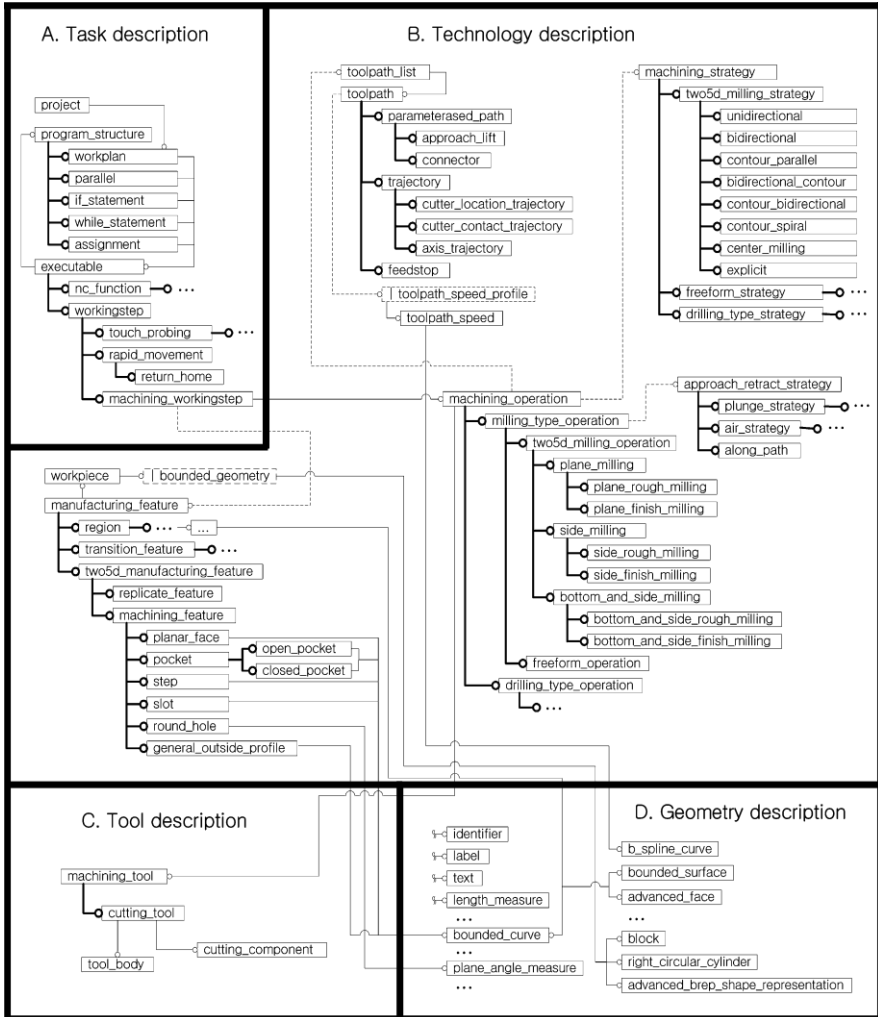


Figura 7 - EXPRESS-G representação do esquema geral da ISO 14649.

Fonte: (SUH, et al., 2003)

Dependendo da forma como a norma (ISO 14649-11) é implementada no CNC, existem três tipos de controles a serem implementados: Tipo 1 (Controle convencional), Tipo 2 (Novo controle) e Tipo 3 (Novo controle inteligente), como mostra a Figura 8.

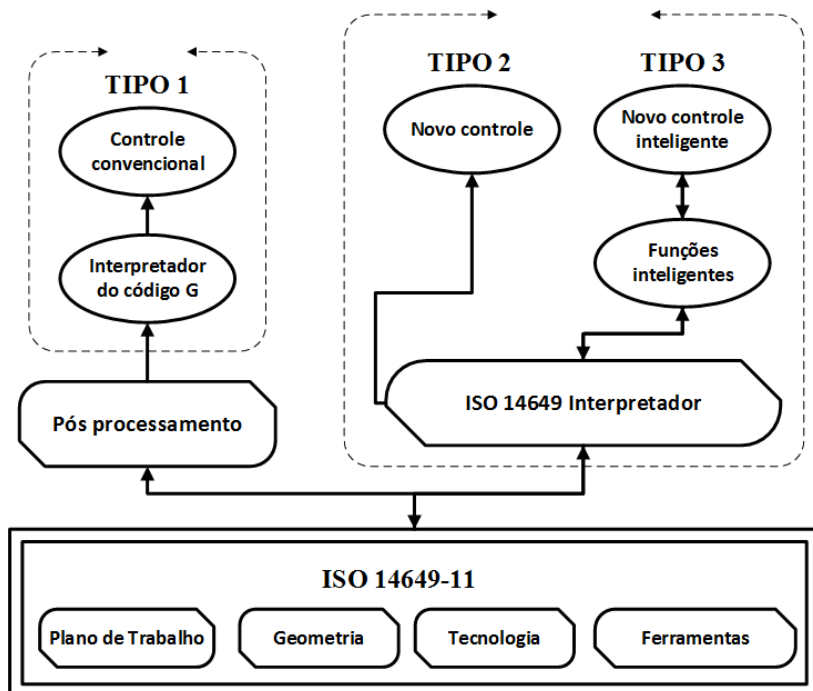


Figura 8 - Três tipos de STEP-CNC.

Fonte: Adaptado de (SUH, *et al.*, 2003)

- 1) O primeiro tipo adiciona a norma (ISO 14649-11) ao controlador convencional através do pós-processamento. Nesse caso, um CNC convencional pode ser usado sem modificações, uma vez que deve pelo menos ser capaz de ler o padrão STEP-NC da norma (ISO 14649-11).
- 2) O segundo tipo, o 'Novo Controle', que tem um interpretador STEP-NC no qual a entidade *workingstep* programada é executada pelo *kernel* do CNC com capacidade de geração do caminho da ferramenta. Este segundo tipo é básico e o movimento é executado

com base na sequência e estratégia de usinagem do programa da peça conforme norma (ISO 14649-11).

- 3) O terceiro tipo, muito mais promissor do que os pós-processadores, é o “Novo Controle Inteligente” no qual o CNC é capaz de realizar tarefas de usinagem inteligente e autônoma com base nas informações da norma (ISO 14649-11). Alguns exemplos das funções inteligentes são as de reconhecimento automático de recurso, sistema automático de anti-colisão, geração do caminho da ferramenta, incluindo o movimento de aproximação e o de retração do eixo árvore, seleção automática da ferramenta, seleção automática das condições de corte, monitoração e reajuste automático das trajetórias da máquina durante o processo de usinagem.

Atualmente existem duas versões do padrão STEP-NC sendo desenvolvidas, que são o ARM (*Application Reference Model*), que corresponde ao padrão STEP-NC propriamente dito norma (ISO 14649-1) e o AIM (*Application Interpreted Model*), que corresponde ao (ISO 10303-1) AP-238 (HASEGAWA, 2010). A principal diferença entre esses dois modelos é o grau em que eles usam os métodos de STEP para a representação e a arquitetura técnica. Ambas as versões podem ser vistas como diferentes métodos de execução de STEP-NC. A indústria tem debatido se deve-se usar AIM ou ARM. Além disso, muitas inconsistências entre as normas devem ser resolvidas para padronizar o modelo de dados STEP-NC. Por conseguinte, uma grande quantidade de atividades continua a ser necessária para viabilizar comercialmente as tecnologias relacionadas ao STEP-NC (YUSRI, 2010).

Alguns dos benefícios da utilização do STEP-NC são listados abaixo (HARBS, 2012).

- Devido à facilidade com que as tarefas são definidas, há uma redução significativa no tempo de planejamento no CAM e na preparação dos dados.
- O tempo de usinagem para lotes pequenos e médios pode ser reduzido devido à otimização inteligente do controlador CNC aderente a STEP-NC.
- Os pós-processadores serão eliminados, uma vez que a interface não requer informações específicas para uma dada máquina.

- As máquinas-ferramenta tornam-se melhor adaptáveis porque o STEP-NC é independente do desenvolvedor de máquina-ferramenta.
- O STEP-NC fornece um modelo de dados completo e estruturado, ligado com a informação geométrica e tecnológica e, deste modo, as informações não são perdidas entre os diferentes estágios do processo.
- Arquivos em formato XML (*Extensible Markup Language*) podem ser utilizados para transferir informações e, em consequência, permitem a manufatura baseada na *web* (ou e-manufatura).
- Descrições de percurso das ferramentas são independentes da geometria da máquina.

Em resumo, a norma (ISO 14649-10) descreve um novo modelo de transferência de dados entre CAD/CAM e os sistemas de máquinas CNC, que substitui a norma (ISO 6983-1).

Porém, para entender o modelo de dados do (ISO 14649-1), deve-se entender primeiramente as limitações que a norma (ISO 6983-1) possui. O código G contém apenas o movimento dos eixos, velocidade, avanço, troca de ferramenta, posição da ferramenta e *On/Off* do refrigerante conforme Figura 9.

Como estas informações que são muito básicas, a compreensão do fluxo operacional fica comprometida, assim como as condições de usinagem e a especificação de ferramentas quando se lê o programa NC de uma peça. Devido a isto, é virtualmente impossível para o controlador CNC executar um controle inteligente e autônomo lidando com erros e casos de emergência (SUH *et al.*, 2008).

Diante das limitações da norma (ISO 6983-1), o padrão STEP-NC contém informações funcionais necessárias conforme a Figura 9, como os dados geométricos, dados das *features* de usinagem e dados do processo.

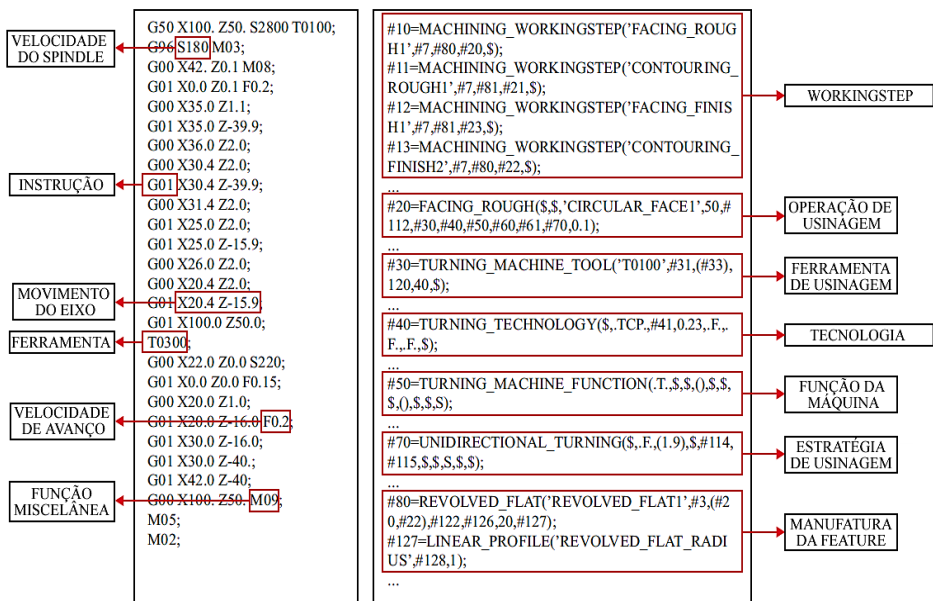


Figura 9 - Código G em comparação ao código STEP-NC.

Fonte: Adaptado de (SUH, *et al.*, 2008)

3.8. INTEGRAÇÃO CAD/CAPP/CAM/ CNC COM STEP-NC

O padrão STEP-NC (ISO 14649-11) fornece um modelo de dados orientado a objetos para CNCs com uma interface de dados estruturados e detalhados que incorporam a programação baseada em *features* a serem usinadas, nos tipos de ferramentas utilizadas, nas operações a serem realizadas (*Workingsteps*) e no plano de processo (*Workplan*). A estrutura de dados STEP-NC usa um modelo orientado a objetos para descrever as entidades, tais como *Workplan*, *Workingstep*, *Features*, operações, ferramentas, bem como todas as entidades geométricas. Desta maneira, cada entidade pertence a uma classe de objetos e deve ser utilizada como tal, onde aponta-se para as partes de cada objeto ao invés de telas em um único registro físico/linha (PEREIRA, 2003).

No STEP-NC, as características das peças geradas no CAD são utilizadas para descrever os volumes a serem removidos na usinagem, garantindo na usinagem o formato da peça que foi desenhada. Estas

características são reconhecidas pelo sistema CAM, que contém a geometria final e as tolerâncias que são utilizadas no processo de usinagem. Em muitos casos os recursos finais utilizados na geometria podem ser usados diretamente, como recursos de compensação do raio de ferramentas e estratégias de usinagem (CHOI *et al.*, 2006).

Estas características baseiam-se na geometria da peça bruta, e a geometria final é derivada das características de concepção da peça. O padrão STEP-NC acrescenta ao CAM, a possibilidade de alteração dos parâmetros de usinagem automaticamente quando as operações e *workingsteps* são especificados pelo engenheiro de projeto (WOSNIK, KRAMER e KLEMM, 2009).

O modelo de dados da Figura 10 contém dados de geometria, de produção, de recursos e do processo de fabricação. Os dados de geometria são tipicamente originados do CAD e são descritos nesta etapa, incluindo todas as informações necessárias para definir a geometria final da peça a ser usinada (SÄÄSKI, SOLONEN e PARO, 2005).

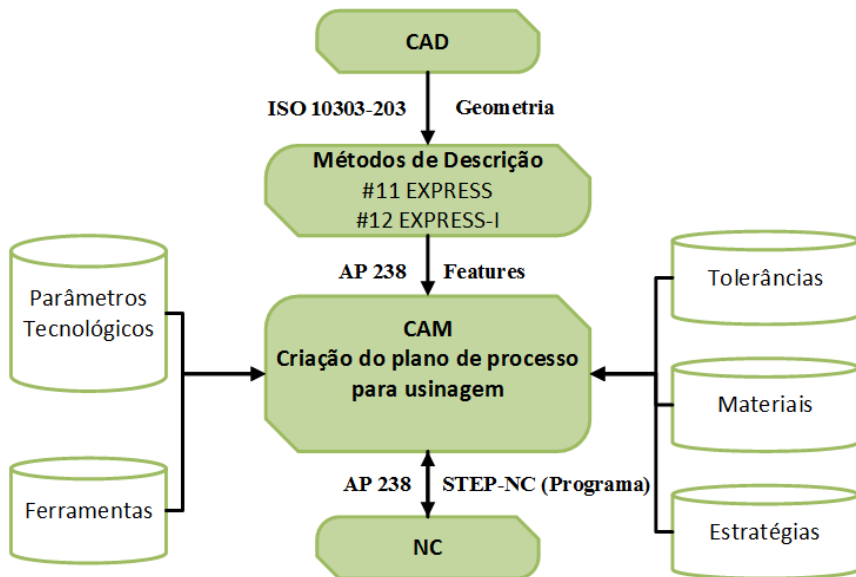


Figura 10 - Descrição geral do modelo de dados e o fluxo de dados.

Fonte: Adaptado de (SÄÄSKI, SOLONEN e PARO, 2005).

O STEP-NC descreve o que deve ser feito nas tarefas de pré-perfuração, perfuração, desbaste, acabamento etc., proporcionando um maior nível de informação sobre as tarefas de usinagem e tecnologia de dados sobre informações geométricas e topológicas, enquanto o código G descreve como fazer. Como resultado, as modificações no chão de fábrica tais como velocidade de corte, velocidade por rotação, avanço por dente, profundidade (corte, trabalho e avanço), permite que uma troca direta de informações entre CAD/CAM e NC possa ser realizada (XU e HE, 2004).

3.9. PROGRAMAÇÃO PARA OPERAÇÃO DE FRESAMENTO NO PADRÃO STEP-NC

A peça da Figura 11 inclui um plano na face superior (*planar face*), uma cavidade retangular (*closed_pocket*) e um furo (*round_hole*). Nesta seção serão explicadas as sequências de usinagem e informações detalhadas sobre uma cavidade retangular e sua operação de usinagem.

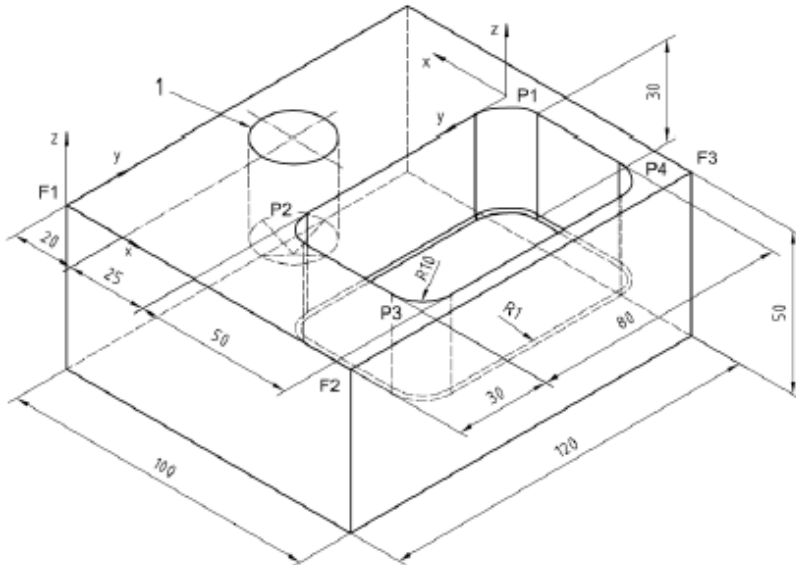


Figura 11 - Exemplo simples de uma parte do programa para teste de fresamento.

Fonte: (ISO 14649-11)

A Figura 12 mostra um exemplo de fresamento contendo a estrutura geral do programa STEP-NC (ISO 14649-11). Observa-se que é apenas um trecho de todo o programa, o qual contém informações que mostram a sequência de usinagem. Cada programa STEP-NC começa com a entidade *project* (#1), que tem como finalidade chamar outras duas importantes entidades: “*workplan*” (plano de trabalho) e “*workpiece*” (peça de trabalho). O *workplan* faz parte das entidades executáveis e pode conter outros *workplans*, assim como um ou vários *workingsteps*, que se apresentam dispostos em uma sequência linear. Os principais objetivos do *project* são definir a sequência dos processos de usinagem usando o atributo *main_workplan* (#2) e definir as informações da peça usando o atributo *workpiece* (#4). Na Figura 12 cinco *machining_workingsteps* são executados sequencialmente. Primeiramente, a operação de acabamento para o *planar_face* na parte superior (#10) é executada, e em seguida as operações de furação (#11) e de mandrilamento (#12) são executadas para o *round_role*. Finalmente ocorre o desbaste (#13), o acabamento (#14) e as operações de *closed_pocket*.

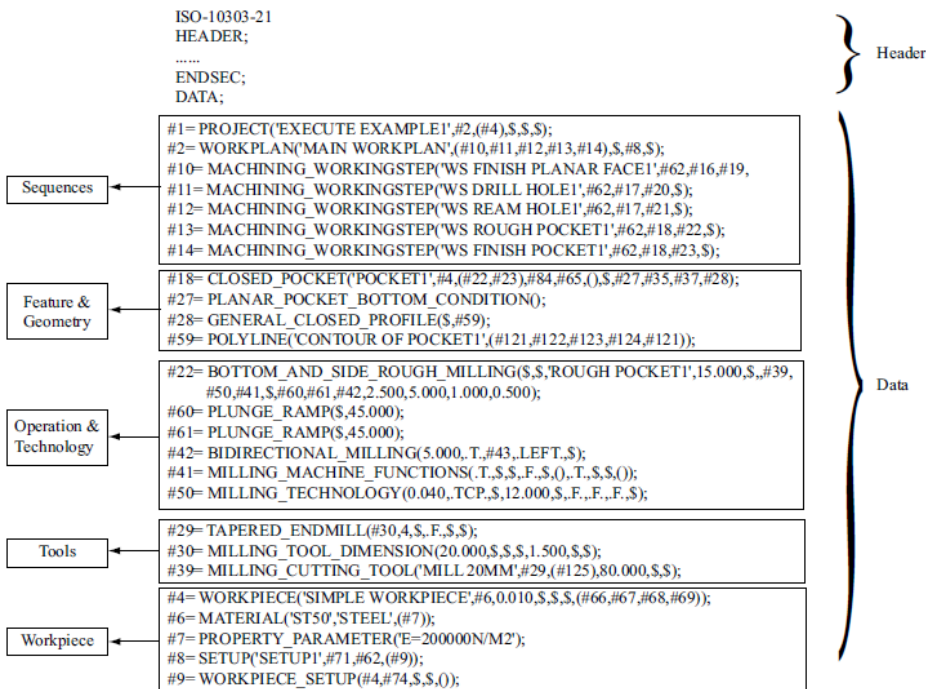


Figura 12 - Parte do programa STEP-NC de um fresamento.

Fonte: (ISO 14649-10)

O trecho "*Feature & Geometry*" contém informações de recursos no programa STEP-NC, especialmente a *feature closed_pocket*. No programa da peça, a parte inferior do *pocket* (cavidade) é definida como *planar_pocket_bottom_condition* (#27). A operação *general_closed_profile* (#28), mais especificamente a *polyline* (#59), é utilizada para o contorno da operação *closed_pocket* (SUH *et al.*, 2008).

A seguir, o programa para a operação de desbaste será explicado. O tipo de usinagem é dado pela entidade *bottom_and_side_rough_milling* (#22), que tem informações de profundidade axial (4,0mm), profundidade radial (3,0mm) e acabamento para a parede lateral (1,0mm) e para o fundo da cavidade (1,0mm), o ponto de partida e o comprimento.

O *machining_strategy* define o método para executar a operação de usinagem. A entidade *bidirectional_milling* (#42) é utilizada no plano de processo, e ela define a direção da máquina, passo ao longo da direção e assim sucessivamente. Se esses valores forem omitidos, o CNC pode decidir esses valores de forma autônoma. A entidade *milling_technology* (#50) define as condições de usinagem, tais como o avanço e a rotação do eixo-árvore. O avanço pode ser definido usando-se os atributos *feedrate* ou *feedrate_per_tooth* (avanço por dente) e a rotação do eixo-árvore pode ser definida usando-se *spindle* ou *cut_speed*. Informações adicionais, tais como o movimento simultâneo do eixo-árvore e de avanço, a substituição do avanço e do eixo-árvore podem ser definidos. Neste exemplo, o *feed_per_tooth* (avanço por dente) é usado para definir alimentação, enquanto o *cut_speed* é usado para definir a velocidade de corte. A entidade *milling_machine_functions* (#41) define os elementos da máquina-ferramenta, tais como a pressão do ar, o fluido de corte que é utilizado na remoção de cavacos durante a usinagem, a remoção dos cavacos, etc. Para o *machining_tool*, *taper_endmill* é utilizado o parâmetro (#29), o qual define o diâmetro (20,0mm), o raio do gume (1,5mm), o comprimento total (80,0mm) e o número de dentes (4).

A informação sobre a matéria-prima da peça é definida no STEP-NC pela entidade *workpiece*. Deve-se mencionar que o código G não fornece informações sobre a peça e, por conseguinte, somente o operador sabe qual é a informação contida na peça e decide as condições de corte para a mesma, considerando as informações geradas no código G. Por outro lado, o STEP-NC inclui as formas inicial e final da peça, o tipo de material da peça a ser usinada, a posição da peça no processo de usinagem, etc. (SUH *et al.*, 2008).

3.10. PROJETOS RELACIONADOS À NORMA STEP-NC (ISO 14649)

A norma ISO14649 possui vários projetos que foram desenvolvidos e construídos para que o CNC pudesse trabalhar neste novo padrão. Alguns deles continuam atualmente ativos em conjunto com o desenvolvimento de outras normas. A seguir são apresentados alguns desses projetos.

O Projeto STEP-NC MACHINE: esse projeto é implementado a partir do antigo projeto SuperModel da empresa de software (STEP Tools). O resultado final é o software STEP MACHINE, que tem sido alvo de provas e melhoras contínuas desde sua concepção até hoje. É de notar que esse software, desde o ano 2000, tem sido utilizado como banco de ensaio STEP-NC em uma série contínua de demonstrações de tecnologia para verificar o padrão ISO10303, implementações da unidade de comandos de máquina STEP-NC, além de proporcionar à indústria da manufatura experiência com a norma STEP-NC.

Em junho de 2003 a STEP Tools disponibilizou um *plugin* para o software CAM MASTERCAM. Esse *plugin* baseia-se no software STEP-NC MACHINE, o qual possui várias versões. Ressalta-se que, na segunda versão, foram construídas as primeiras interfaces de controle com base em STEP-NC para máquinas CNC Fanuc e Siemens que originalmente usavam código G. Os resultados obtidos foram mostrados no primeiro fórum de teste industrial em 2005. Desde 2010 esse software está na oitava versão (STEP-NC MACHINE v8) (STEP Tools).

O Projeto ISO14649 ToolKit: esse projeto foi iniciado em dezembro de 2008 e concluído em fevereiro de 2009 por Mark Pictor e Thomas Kramer, ambos programadores de software de manufatura aberta (ISO 14649-TOOLKIT, 2009). Eles criaram um conjunto de ferramentas (*toolkit*) para STEP-NC baseado nos códigos: sebnf2pars (SEBNF2PARS, 2009). O sebnf2pars é uma ferramenta para a construção de analisadores de arquivos de STEP Part 21 e do código do (NIST SCL, 1999), da norma ISO14649 partes 10, 11 e 111.

Entre as ferramentas incluídas no software do *toolkit*, existe uma que permite analisar código no formato STEP Part 21 ou “.p21” para saber se a sintaxe está correta. Também há uma ferramenta para interpretar e imprimir código NC para usinagem de três eixos.

O *toolkit* foi programado na linguagem C++, e admite a modificação do seu código fonte para acoplá-lo às necessidades de

qualquer outro projeto com relação a controladores NC como, por exemplo, o EMC2.

O Projeto STEP Modeler UFSC/GRIMA: esse projeto de doutorado foi desenvolvido na Universidade Federal de Santa Catarina (UFSC) em conjunto com Grupo de Integração da manufatura (GRIMA) com o nome: “UM SISTEMA PARA O PROJETO E FABRICAÇÃO DE PEÇAS MECÂNICAS A DISTÂNCIA VIA INTERNET ADERENTE À NORMA ISO 14649 (STEP-NC)” (BENAVENTE, 2011). O resultado final foi o software chamado “STEP Modeler”.

O STEP Modeler aproveita a tecnologia da internet introduzindo a e-manufatura. A e-manufatura consiste em um fluxo de dados compatíveis e sem brechas na cadeia CAD-CAPP-CAM-CNC (BENAVENTE, 2011). Este software usa o padrão ISO14649 voltado para fresamento (partes 10, 11 e 111). Apesar do padrão ISO14649 conter muitas informações para usinagem de peças prismáticas e outras, só foram disponibilizadas algumas das *features* descritas na norma ISO14649 parte 11.

Com o STEP Modeler é possível visualizar uma peça prismática junto com *features* adicionadas, isso com o fim de ter uma ideia da peça acabada além de se verificar se a peça criada é válida desde o ponto de vista de fabricabilidade com as ferramentas disponíveis. Por último o sistema gera automaticamente o arquivo físico no formato p21.

Concluído parcialmente em dezembro de 2011, atualmente apresentam-se modificações no seu código, melhorando-se constantemente com relação à geração do código no formato físico STEP part 21 ou “.p21” usando a norma ISO14649 de 2002.

É importante ressaltar o funcionamento deste software, pois foi usado neste projeto para gerar o arquivo físico de uma peça CAD projetada, exportando-o no formato de saída “STP” ou “.p21” (padrão ISO 10303 part 21).

O Projeto Controlador CNC aderente a ISO 14649 com ferramentas de software para construção e integração de uma arquitetura de controle baseada na IEC 61499:

Neste trabalho foi desenvolvimento de um controlador CNC aderente às normas ISO 14649 (STEP-NC) e IEC 61499 (FB), chamado de CNC-C2 (CNC Compliant to STEP-NC and Compliant to IEC 61499).

O arquivo STEP-NC aplicado no controlador que foi desenvolvido, atua como entrada de informações, sem o uso do código

G/M, tendo uma arquitetura para comando da máquina o uso de FBs (*function blocks*).

O protótipo desenvolvido é constituído de uma fresadora 2.5D, acionada por um conjunto de três servoacionamentos com CLPs integrados e ferramentas de software para compilação do arquivo STEP-NC, visualização, edição e execução de FBs e rede de FBs e de uma biblioteca de modelos IEC 61499.

Diante destes trabalhos, no próximo capítulo foi utilizado parcialmente a metodologia baseada nos procedimentos de metodologia e de pesquisa científica descritas por (MENEZES e SILVA, 2005).

4. MATERIAIS E MÉTODOS

Desde 2007 o Grupo de Integração da Manufatura (GRIMA) do Departamento de Engenharia Mecânica da Universidade Federal de Santa Catarina vem desenvolvendo um sistema CAD/CAPP/CAM para o projeto e fabricação de peças prismáticas via Internet, o qual contém importantes aspectos de padronização, de acordo com a norma (ISO 14649-11) - STEP-NC (BENAVENTE, 2011); (BENAVENTE *et al.*, 2013). Este sistema CAD/CAPP/CAM gera o arquivo físico no formato STEP-NC e disponibiliza-o para o usuário remoto via Internet, além de propiciar a visualização e a simulação da trajetória das ferramentas. O sistema computacional, denominado *STEP Modeler*, pode ser executado na página http://www.grima.ufsc.br/stepnc_project/. Entretanto, as máquinas CNC disponíveis no Departamento de Engenharia Mecânica da UFSC não são aderentes ao formato STEP-NC.

Para fins de pesquisa e implementação do padrão STEP-NC norma (ISO 14649-11) em uma máquina CNC, em 2012, a UFSC realizou a compra de uma fresadora CNC do tipo pórtico (*router*) de três eixos, a qual é mostrada na Figura 13 e está localizada no Laboratório de Hardware (LHW), Departamento de Engenharia Mecânica.

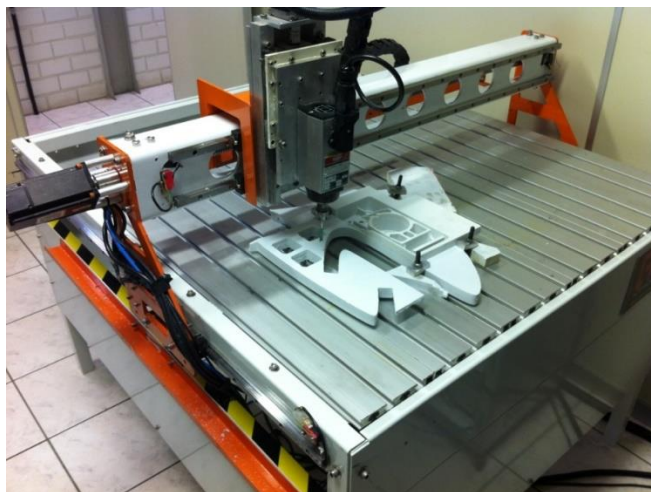


Figura 13 - Fresadora *Router* CNC.

Fonte: Laboratório de Hardware II.

A Figura 14 mostra as etapas para a fabricação de peças na fresadora CNC utilizando-se o código G. O fluxograma mostra o processo de fabricação desde a definição do desenho da peça no *software* CAD (SolidWorks), que é um *software* de modelagem de sólidos, paramétrica, gerando-se o arquivo no formato SLDPRT (proprietário do SolidWorks) até a importação para o *software* CAM (EdgeCAM), permitindo criar estratégias de usinagem para máquinas CNC possibilitando a geração do código G a partir de modelos CAD. O arquivo em código G é visualizado no *software* CNC (CNC USB Controller), que é o responsável por enviar o código G para a placa (Planet CNC) responsável pela interpretação das trajetórias e envio dos sinais de pulso e direção para os *servo drivers* de acionamento dos eixos X e Y da máquina modelo ASDA-B2 da fabricante Delta Electronics, assim como também para o *driver* de acionamento do eixo Z (cabecote porta-ferramenta) modelo 57BYGH115-003 do fabricante Wantai Motor.

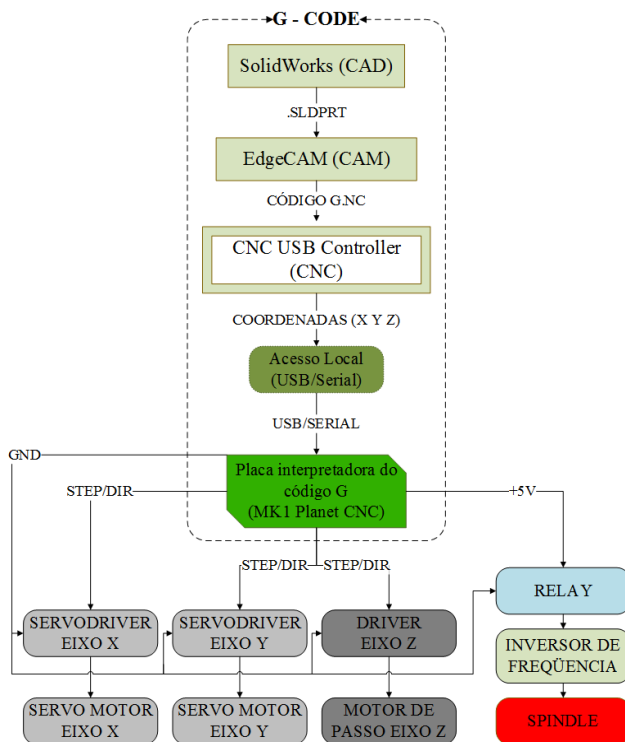


Figura 14 - Fluxograma do processo de fabricação atual na fresadora CNC.

Fonte: Elaborado pelo autor.

Os *softwares* CAD (SolidWorks) e CAM (EdgeCAM) foram utilizados com licença de estudante, já o *software* CNC (CNC USB Controller) possui licença registrada. Um exemplo da interface do programa CNC USB Controller é mostrado na Figura 15. Estes *softwares* são responsáveis pela integração CAD/CAM/CNC para a fabricação de peças na fresadora CNC usando-se o código G.

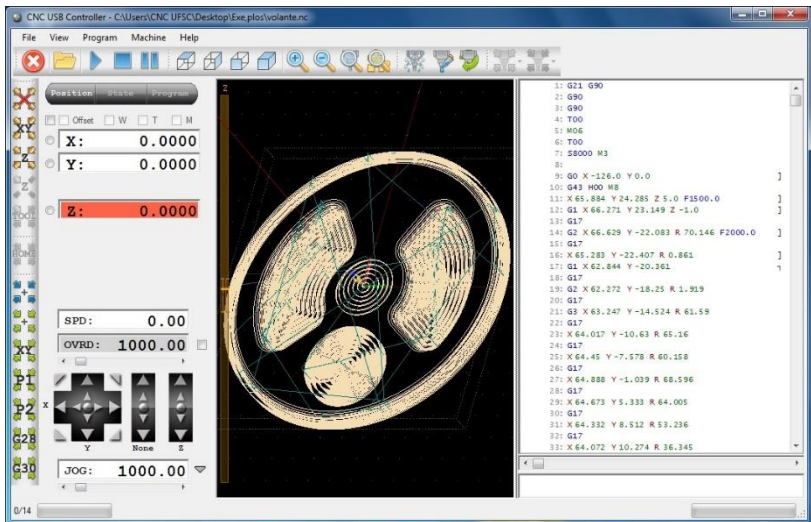


Figura 15 - CNC USB Controller.
Fonte: Software CNC USB Controller.

Já o *hardware* MK1 é mostrado na Figura 16, e a configuração de cada conector é explicado conforme o APÊNDICE A. Ele é utilizado para a transmissão dos sinais de pulso e direção para a movimentação dos três eixos (X, Y e Z) da fresadora CNC.

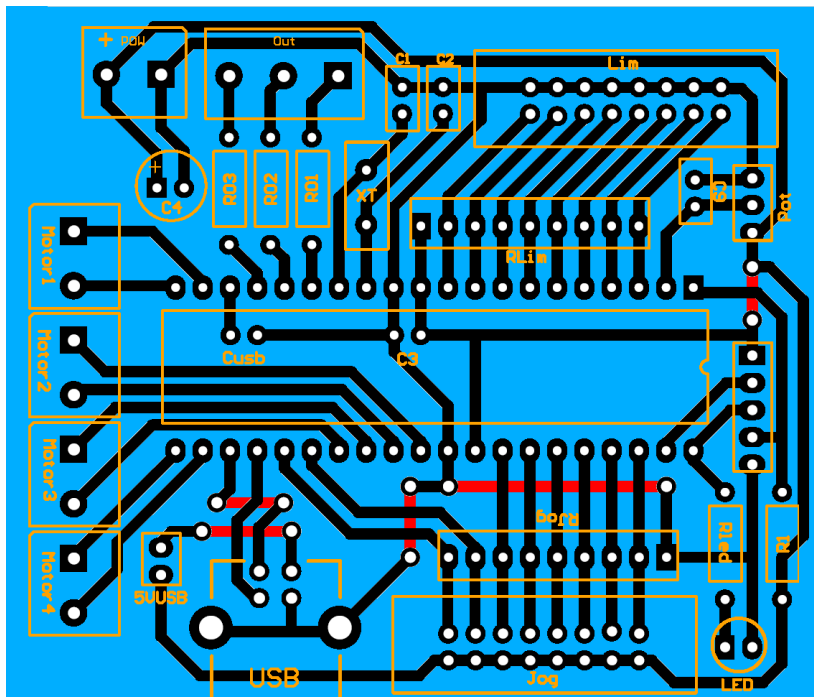


Figura 16 - CNC USB Controller MK1.

Fonte: <http://www.planet-cnc.com/>

O *hardware* da placa controladora MK1 (Planet CNC) controla até 4 eixos e requer uma fonte de 5V de alimentação. Outras características da placa são apresentadas abaixo:

- Frequência máxima de 25 kHz
- 3 saídas digitais para (fluido de corte, ar comprimido e eixo-árvore);
- 12 μ s (largura de pulso);
- Entrada *jog* manual ativa para todos os eixos;
- Limite de entrada para todos os eixos;
- Controla dispositivos externos com o protocolo I2C.

Foi desenvolvido um interpretador de arquivo STEP-NC para a geração de trajetória de ferramentas e, por fim, foi implementada a operação remota da máquina via Internet, conforme descrito nas seções 4.1, 4.2, 4.3 e 4.4.

4.1. CONVERSÃO DA FRESADORA CNC ATUALMENTE COM CÓDIGO G PARA O PADRÃO STEP-NC

Na fresadora CNC a placa responsável pelos sinais que habilitam o eixo-árvore, envio dos sinais de pulso, direção e GND para os *servo drivers* dos eixos X e Y, e *driver* do eixo Z para a movimentação da máquina é a placa controladora MK1 da Planet CNC conforme mostrado no fluxograma da Figura 17.

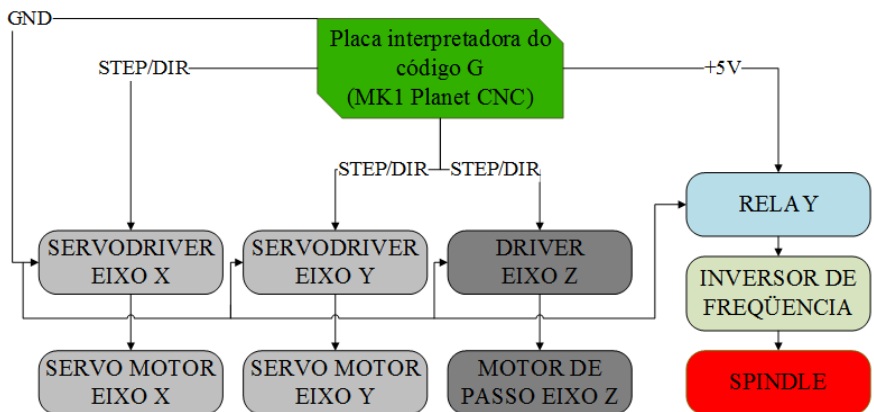


Figura 17 - Caminho percorrido do pulso de 5V para o eixo-árvore (pulso e direção), para os *servo drivers* dos eixos X e Y, e *driver* do eixo Z da fresadora CNC.

Fonte: Elaborado pelo autor.

Para realizar a conversão da fresadora CNC foi necessário encontrar uma forma de separar os sinais que habilitam não só o acionamento do eixo-árvore, mas também os sinais de pulso/direção e GND advindos do controlador MK1, responsável pela leitura e interpretação do código G na máquina CNC. Esta separação dos sinais foi fundamental para que o operador da máquina pudesse trabalhar tanto com o código G quanto o padrão STEP-NC.

A solução encontrada foi a instalação de uma placa com 8 relés conforme a Figura 18, e sua instalação foi relativamente simples no painel de controle da máquina sem alterar a parte elétrica. Foi necessária uma alimentação externa de 5V, a qual pode ser alimentada pela própria placa MK1 da empresa Planet CNC. Utilizou-se um microcontrolador com saídas de 5V como fonte de alimentação para *hardware* externo. Cada relé tem a função de chaveamento e divisão de sinais, podendo conter 3 sinais conectados em cada um dos seus 8 relés, em um total de 24 sinais.



Figura 18 - Placa seletora Relé de 8 canais.

Fonte: Elaborado pelo autor.

Ela é capaz de controlar vários aparelhos porque pode ser controlada diretamente por microcontroladores como *Raspberry Pi*, *Arduino*, 8051, AVR, PIC, DSP, ARM, MSP430 e lógica TTL.

Cada um dos 8 canais recebe uma tensão de 5V e precisa de uma corrente de 15 a 20mA, além de suportarem alta corrente para AC250V 10A e DC30V 10A. Esta placa possui em cada relé 8 *LEDs* para indicação do status de ativado em 5V ou desativado em 0V, além do contato NF (normalmente fechado), NA (normalmente aberto) e comum para todos. Todos os 8 relés precisam em cada canal, uma tensão de alimentação de 5V para que eles comutem de NF para NA, proporcionando o redirecionamento do caminho do pulso e direção na máquina CNC, o que será explicado na próxima seção.

4.2. CONFIGURAÇÃO DO HARDWARE PARA QUE O CNC DA MÁQUINA TRABALHE DE FORMA SELETIVA (CÓDIGO G OU STEP-NC)

A utilização da placa de 8 relés foi responsável por realizar a alteração de forma seletiva dos padrões (código G e STEP-NC) na máquina fresadora CNC. A mesma possui em seu painel de controle o microcontrolador MK1 responsável pela interpretação e transmissão das trajetórias para a máquina CNC, porém esta placa não permite suporte e nem integração de uma programação para interpretar o padrão STEP-NC por ela ser *closed source*.

Portanto, para que o padrão STEP-NC pudesse ser lido e interpretado e houvesse a transmissão das trajetórias da ferramenta para a fresadora CNC, foi utilizado um microcontrolador *open source* que permitisse a programação e adaptação da máquina juntamente com o microcontrolador MK1.

Foi utilizado o microcontrolador Arduino Mega2560 para desempenhar a função de execução e transmissão das trajetórias do padrão STEP-NC para a máquina CNC, além dos sinais que habilitam o eixo-árvore e sinais de pulso e direção para os eixos X e Y que são movimentados por servomotores, e também para o eixo Z que é movimentado por um motor de passo.

O *hardware* utilizado para transmissão de dados, Arduino, possui uma plataforma de desenvolvimento *open source* para a adequação do padrão STEP-NC à fresadora CNC.

Ele surgiu de um projeto acadêmico sendo associado à filosofia de “*Physical Computing*”, unindo sistemas físicos utilizando *software* e *hardware* que respondem a entradas vindas de *software*.

O motivo da escolha do Arduino deveu-se ao fato do seu sucesso, uma vez que atualmente encontra-se algumas fresadoras CNC em funcionamento utilizando o Arduino como microcontrolador na máquina já contendo a programação para o código G.

Isso é possível devido à programação do GRBL (<https://github.com/grbl/grbl>), que é um programa em C projetado para interpretar o código G e enviá-lo para máquinas CNC, como fresadoras.

O sucesso deste microcontrolador está relacionado ao seu baixo custo e às suas finalidades. Sua simplicidade na utilização e a possibilidade de utilização em vários sistemas operacionais como em Windows, Macintosh OS e Linux. Existem três linhas principais de modelos de Arduino (<http://www.arduino.cc/>):

- Os de "pequeno porte": NANO
- Os de "médio porte": Duemilanove, UNO
- Os de "grande porte": Mega1280, Mega2560

O Arduino Duemilanove e o UNO possuem um número de portas de entrada e saída menor do que os modelos Mega1280 e Mega2560, além de terem menor memória *flash* (responsável pela gravação do código). Entretanto, ele é mais leve e compacto do que os modelos MEGA. Outra grande diferença é que o *chip* Atmega é DIP (*dual-inline-package*) no Duemilanove/UNO, e isso permite retirar este *chip* e trocá-lo por outro em caso de danos ou para fazer um projeto sem o uso da plataforma Arduino. No Arduino Mega o Atmega é SMD (*surface mounted*), não permitindo facilmente a remoção e troca do microcontrolador. A escolha do Arduino Mega2560 da empresa Atmel para este projeto, mostrado na Figura 19, foi devido à quantidade de entradas/saídas, incluindo analógicas, e por possuírem maior capacidade de memória do que os modelos Duemilanove e o UNO (<http://www.arduino.cc/>).



Figura 19 - Arduino Mega2560.

Fonte: Elaborado pelo autor.

Uma grande vantagem é que o Arduino possui uma plataforma de *hardware* já modelada e reconfigurável via *firmware*, dispensando em muitos casos os conhecimentos de eletrônica digital e arquitetura de microcontroladores em relação ao controlador da placa. Bastando apenas um simples conhecimento em algoritmos para realizar uma aplicação.

Porém a grande desvantagem é ao mesmo tempo essa, pois o programador fica preso ao sistema Arduino que não é portátil, e não te dá conhecimentos na arquitetura de um microcontrolador sendo difícil para o programador migrar para outra plataforma.

Uma vez definido o *hardware* Arduino Mega2560 contendo o microcontrolador Atmega2560, foi realizada a programação para leitura do arquivo padrão STEP-NC e o envio de trajetórias e sinais para a máquina.

Os microcontroladores deverão trabalhar separadamente porque cada plataforma, placa (MK1 e Arduino Mega2560) envia sinais (STEP, DIR e 0V) para cada eixo (X, Y e Z) assim como também sinais de (+5V e 0V) para o relé que aciona o inversor de frequência para funcionamento do eixo árvore. Estes sinais não podem ser conectados juntos e sim separados para que não haja problema de sinais de um microcontrolador sendo enviado para o outro microcontrolador.

Para que esta separação ocorra, a placa de relés com 8 relés precisou ser configurada para poder atender as principais necessidades propostas nesta dissertação. Basicamente, como cada um dos 8 relés possui contatos (NF, NA e comum), foram separados todos os sinais (pulso, direção, GND) de cada um dos eixos da fresadora, assim como o sinal +5V que fecha um circuito no inversor de frequência responsável por habilitar o eixo-árvore.

Uma forma encontrada para mostrar como foi realizada a separação dos sinais tanto vindos do microcontrolador MK1 (código G) como do microcontrolador Arduino Mega2560 (STEP-NC) é apresentada na Figura 20.

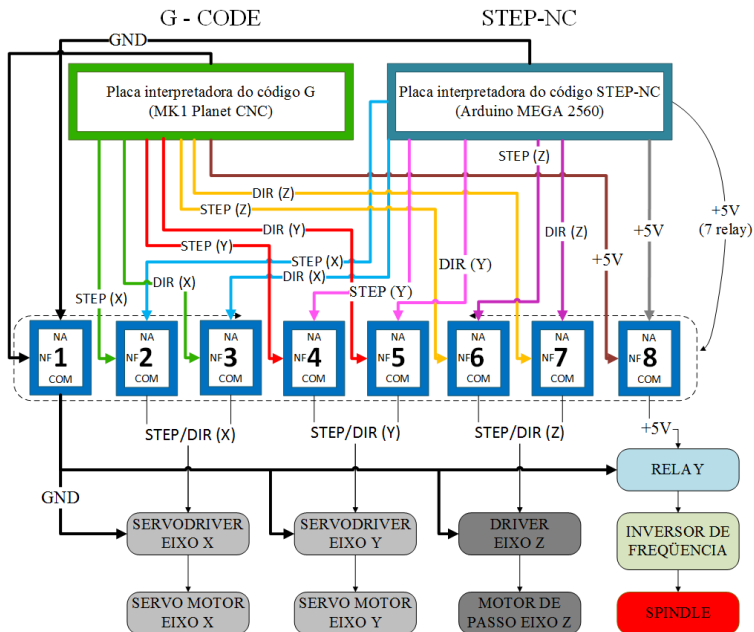


Figura 20 - Fluxograma de separação dos sinais (pulso, direção, GND e do eixo-árvore) na placa 8 relés.

Fonte: Elaborado pelo autor.

Para que os sinais de pulso, direção e GND de cada um dos eixos da fresadora, bem como o sinal que habilita o eixo-árvore da placa MK1 passassem pela placa de relés com 8 relés, foram utilizados cada um dos 7 contatos NF de cada relé da placa para transmissão dos sinais. Estes sinais passam pelo contato COM (comum), o qual é recebido pelos servomotores dos eixos X e Y, e pelo *driver* do motor de passo do eixo Z. Para os mesmos sinais que vêm da placa Arduíno Mega2560, foram utilizados cada um dos 7 contatos NA de cada relé da placa para transmissão de sinais. Quem habilita a comutação de NF para NA é um comando do *software* STEP-NC Controller que será explicado no próximo capítulo, o qual foi desenvolvido na linguagem Java para controlar a fresadora CNC no padrão STEP-NC. Este *software* envia um comando para o microcontrolador Atmega2560, o qual em sua programação foi definida uma saída de +5V que habilita 7 dos 8 relés da placa de transmissão de sinais. Para o acionamento do eixo-árvore, o sinal de +5V advindo das placas MK1 ou Arduíno Mega2560 aciona um outro

relé que habilita o contato NA para que este sinal envie um pulso para que o inversor de frequência ligue o eixo-árvore.

4.3. INTERPRETADOR DE ARQUIVO STEP-NC PARA GERAÇÃO DE TRAJETÓRIA DE FERRAMENTAS

Para interpretar o arquivo p21 do padrão STEP-NC, foi preciso utilizar o sistema CAD/CAPP/CAM denominado STEP Modeler, que pode ser executado na página http://www.grima.ufsc.br/stepnc_project/ (BENAVENTE, 2011), (BENAVENTE *et al.*, 2013). Esse sistema permite a modelagem de uma peça prismática via Web baseada em *features* de usinagem e permite ao usuário, gerar o arquivo físico STEP-NC aonde disponibiliza ao usuário o acesso remoto via Internet, além de proporcionar a visualização e a simulação da trajetória das ferramentas.

Mencionado na seção anterior, o arquivo GRBL é uma alternativa de baixo custo baseado em porta paralela para controle de movimentos para um processo de usinagem em máquinas CNC. Ele pode ser executado em Arduino (Duemillanove / Uno e Mega), desde que ele suporte um *chip* Atmega328 ou um Atmega2560.

O controlador é escrito na linguagem de programação C e é altamente otimizado utilizando todos os recursos inteligentes dos AVR-*chips* para conseguir sincronismo preciso e operação assíncrona. O AVR é uma arquitetura de microcontrolador RISC (*Reduced Instruction Set Computing*) de 8 bits em um *chip* modificado que foi desenvolvido pela Atmel. Ele é capaz de manter até 30kHz de estabilidade e pulsos de controle livres.

Por ser compatível com o código G, ele realiza usinagem na forma de arcos, círculos e movimento helicoidal que são suportados por este controlador, bem como outros comandos básicos funcionais do código G. Algumas funções e variáveis não são suportadas atualmente, mas podem ser incluídas em futuras versões na forma de um pré-processador. Além disso, ele inclui um gerenciamento da aceleração da movimentação da ferramenta, já prevendo as próximas trajetórias de acordo com as linhas subsequentes do arquivo em código G (chama-se a isto “*look-ahead*”). O controlador pode ler até 18 movimentos à frente da sequência atual de movimentos, planejando suas velocidades à frente para manter uma aceleração e curvas suaves para evitar que a máquina sofra algum tipo de movimentação brusca comprometendo o desempenho físico da máquina CNC.

O GRBL proporciona uma programação livre, podendo ser distribuída de maneira a ser útil para projetos de pesquisa, mas sem qualquer garantia de que o mesmo funcionará imediatamente sem qualquer modificação no programa. Dos seus arquivos, há bibliotecas no formato “.h” bem como arquivos no formato “.c”, conforme mostrado na Tabela 1.

Tabela 1 - Bibliotecas e arquivos de programação do programa GRBL.

| Bibliotecas “.h” | Arquivos “.c” |
|-------------------------|----------------------|
| config.h | |
| coolant_control.h | coolant_control.c |
| defaults.h | |
| EEPROM.h | EEPROM.c |
| gcode.h | gcode.c |
| limits.h | limits.c |
| | main.c |
| motion_control.h | motion_control.c |
| nuts_bolts.h | nuts_bolts.c |
| pin_map.h | |
| planner.h | planner.c |
| print.h | print.c |
| protocol.h | protocol.c |
| report.h | report.c |
| serial.h | serial.c |
| settings.h | settings.c |
| spindle_control.h | spindle_control.c |
| stepper.h | stepper.c |

Fonte: Elaborado pelo autor.

4.3.1. Análise do código G no programa GRBL

Para compreensão do caminho percorrido correspondendo aos sinais de pulso e direção de cada um dos três eixos da fresadora CNC, foram analisados os arquivos (*serial.c*, *protocol.c*, *gcode.c* e *stepper.c*) do código programado GRBL. O caminho percorrido pelo código G no programa GRBL se inicia pelo arquivo *serial.c*, mais precisamente na interrupção ISR (USUARIO_RX_vect), o qual recebe os dados serialmente *byte* por *byte*. O registrador que guarda o *byte* recebido é a

UDRO, e os *bytes* recebidos são armazenados em uma fila chamada *rx_buffer*. Em paralelo a este caminho, a função *serial_read* é a responsável por ler do *buffer* os *bytes* armazenados e retornar o valor do *byte* lido. O arquivo *protocol.c* é responsável pelo pré-processamento das linhas, executando uma linha de formato de entrada para o *protocol_process*. Ele também executa comandos internos no GRBL como configurações, início do ciclo de *homing*, e alteração dos estados de comutação. Isso difere do módulo de comando em tempo real de execução no GRBL quando o mesmo está pronto para executar a próxima linha durante um ciclo, e nesse caso os interruptores apagam as linhas dos blocos de execução. Esta opção só afeta as linhas que serão processadas mais tarde, não necessariamente em tempo real durante um ciclo, uma vez que existem movimentos já armazenados no *buffer*. No entanto, esse "atraso" não é um problema, uma vez que esses comandos não são normalmente usados durante um ciclo.

No arquivo *gcode.c* a função *gc_execute_line* é a responsável por transformar todas as linhas de código G em parâmetros internos. No arquivo *protocol.c* o *protocol_process* chama a função *serial_read*, que lê o *byte* de uma fila e o retorna. Estes são organizados pelo *protocol_process* em uma matriz chamada *line*. O *protocol_process* envia a *line* para o *protocol_execute_line*, que por sua vez verifica se a linha recebida não é um parâmetro interno: caso não seja, a linha é enviada para a função *gc_execute_line* onde o código G é interpretado. Após o processamento, o arquivo *stepper.c* tem como finalidade enviar os sinais de pulso e direção para os eixos X, Y e Z.

4.3.2. Software desenvolvido para o controle da fresadora CNC para a envio do código STEP-NC

O *software* denominado STEP-NC Controller foi desenvolvido no ambiente Eclipse usando o padrão de programação orientado a objetos. O Eclipse possui licença EPL (*Eclipse Public License*) e é um IDE (*Integrated Development Environment*) para desenvolvimento na linguagem Java que suporta diversas outras linguagens com o auxílio de *plug-ins*, como por exemplo: C/C++, PHP, *ColdFusion*, *Python*, *Scala* e a plataforma *Android*. Este ambiente disponibiliza uma biblioteca gráfica amigável para orientação de desenvolvimento e suporte ao desenvolvedor, com centenas de *plug-ins* que procuram atender as diferentes necessidades de diferentes programadores. O STEP-NC Controller foi projetado não só para enviar o arquivo no padrão STEP-

NC, como também para controlar os movimentos dos eixos X, Y e Z, bem como o acionamento da ferramenta de corte na fresadora CNC. Ele pode ser incluso no *software* STEP Modeler para que o mesmo contenha as atividades desde o projeto do produto até a fabricação da peça em uma máquina totalmente aderente a STEP-NC. O STEP-NC Controller disponibiliza para o usuário uma interface de simples operação para o controle da máquina CNC. Ele passou por uma série de versões até chegar ao estado atual na versão 2.3, tendo sido otimizado para que o Arduino receba, interprete e execute o programa GRBL modificado para o padrão STEP-NC na fresadora CNC.

O *software* STEP-NC Controller é o responsável por executar o arquivo p21 mediante a eventos e trabalha da seguinte maneira: quando um botão é clicado, uma série de eventos ligados àquele botão são disparados, executando-se diversas *threads* (isto é, linhas de execução em um *software*) que são criadas dependendo do botão a ser clicado. Para montar a interface foi utilizada uma extensão do *software* Eclipse que permite uma fácil criação e edição de janelas, usando primariamente as bibliotecas SWT e a *Swing* da linguagem Java para *webcam*. As *threads* utilizadas na interface são mencionadas abaixo para melhor compreensão e detalhamento da interface:

- Janela/Interface: *Thread* principal que renderiza a janela e dispara eventos relacionados aos botões. A janela aberta pelo *Browser* (por exemplo, Google Chrome) na hora se selecionar um arquivo de texto também executa nessa *thread*, impossibilitando ações na janela principal até que esta é fechada.
- Portal Serial USB: Quando conectado ao Arduino localmente, essa *thread* é responsável por receber mensagens advindas do Arduino e enviá-las às classes destino.
- Leitor de arquivo: Quando enviando o arquivo de texto STEP-NC, essa *thread* é responsável por ler o arquivo de texto e responder ao Arduino durante a execução.
- Portal Serial de Rede: Quando conectado via rede, essa *Thread* é responsável por receber mensagens advindas do servidor.
- *Status Update*: comunica com o Arduino para obter o *status* da máquina.
- *Webcam* Janela: responsável por renderizar as imagens da *webcam*.
- *Webcam client*: recebe informações da *webcam* advindas do servidor.

Quando o STEP-NC Controller é executado, uma janela é mostrada na tela, a qual é a janela principal do programa (Figura 21), mediante a qual podem ser acessadas as demais janelas como: *Serial Connection*, *File*, *Network*, *Machine Status*, *Control* e *Console*.

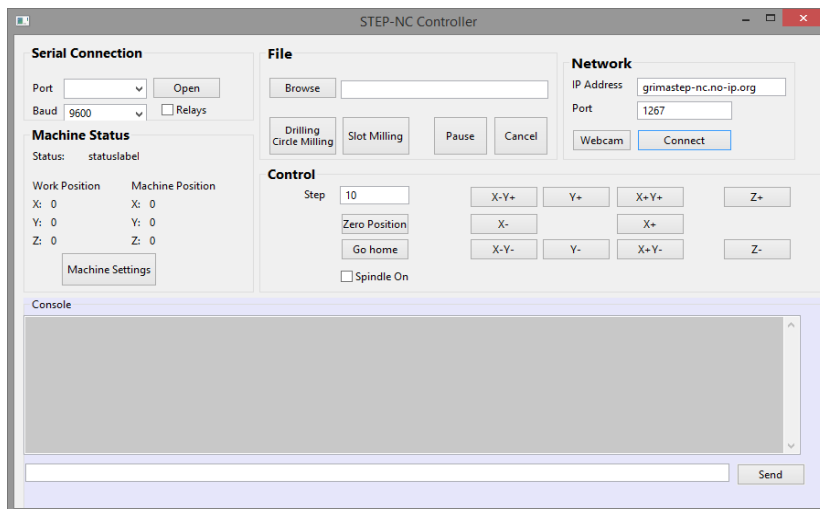


Figura 21 - Interface do software STEP-NC Controller.

Fonte: Elaborado pelo autor.

Para facilitar o uso e identificação das funções do STEP-NC Controller, foram adotadas divisões na interface contendo classes que foram desenvolvidas no ambiente Eclipse para que o usuário pudesse ter maior facilidade de compreensão do *software* de controle da fresadora CNC. A seguir descreve-se cada uma delas:

- ***Serial Connection*** (Figura 22): é uma classe que foi criada para estabelecer a conexão entre a interface do STEP-NC Controller e o Arduino Mega2560 via USB (*Universal Serial Bus*). Uma vez selecionada a porta COM e a *Baud*, que é a velocidade de transmissão de dados entre a interface e o Arduino, o usuário ao pressionar o botão *Open* estabelece uma conexão via USB para troca de dados entre *software* STEP-NC Controller e o Arduino Mega2560. O ícone “Relays” tem como função realizar o chaveamento de 7 dos 8 relés da placa de separação

de sinais. Portanto, o *software* STEP-NC Controller é o responsável pelo envio do pulso/sinal de +5V para a placa de relés com 8 relés o qual comuta de NF para NA separando os sinais de pulso, direção e GND da placa MK1, deixando o caminho livre para os sinais advindos do Arduino Mega2560.



Figura 22 - Serial connection.

Fonte: Elaborado pelo autor.

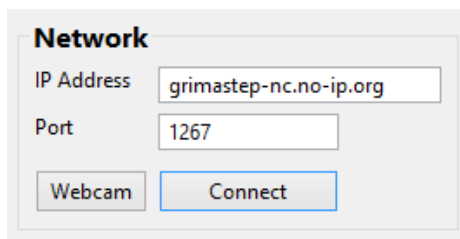
- **File** (Figura 23): é uma classe que possui os botões: *Browse* para busca no computador do arquivo.txt contendo as instruções do padrão STEP-NC, o botão *Drilling/Circle Milling* é responsável por enviar para o Arduino programas de furação e fresamento circular. O botão *Slot Milling* é o responsável pelo envio do programa para fresamento no modo ranhura. Já os botões *Pause* e *Cancel* são responsáveis pela parada momentânea da máquina e parada total.



Figura 23 - File.

Fonte: Elaborado pelo autor.

- **Network** (Figura 24): é a classe responsável pela conexão remota via *internet* do *software* STEP-NC Controller com o Arduino Mega2560. Na seção *IP Address*, o usuário tem liberdade para acessar remotamente a fresadora CNC tanto via link (grimastep-nc.no-ip.org) ou via IP do servidor da UFSC (150.162.27.108); o botão *Connect* estabelece a conexão via Internet. Os botão *Webcam* é o responsável por mostrar a imagem da fresadora CNC em tempo real, tanto pelo acesso local quanto pelo acesso remoto.



Network

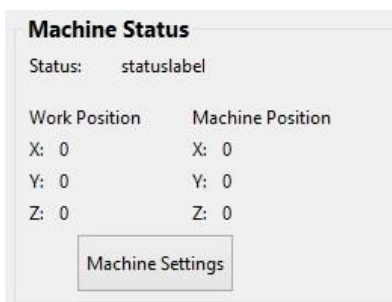
IP Address

Port

Figura 24 - Network.

Fonte: Elaborado pelo autor.

- **Machine Status** (Figura 25): é a classe que mostra para o usuário as coordenadas da máquina nos eixos X, Y e Z, além de possuir o botão *Machine Settings*, mediante o qual o usuário poderá ter acesso às configurações globais do GRBL modificado e adequado ao padrão STEP-NC para armazenar ou alterar valores, definindo assim novos parâmetros à fresadora CNC.



Machine Status

Status: statuslabel

| Work Position | Machine Position |
|---------------|------------------|
| X: 0 | X: 0 |
| Y: 0 | Y: 0 |
| Z: 0 | Z: 0 |

Figura 25 - Machine Status.

Fonte: Elaborado pelo autor.

- **Machine Settings** (Figura 26): cada ícone da aba apresenta as funções e entradas de novos parâmetros para configuração da fresadora CNC visando atender as necessidades dependendo do projeto ou usinagem a ser implementada.

| Machine Settings | |
|-------------------------------------|--------|
| x, step/mm | 100.0 |
| y, step/mm | 100.0 |
| z, step/mm | 80.0 |
| step pulse, usec | 10.0 |
| default feed, mm/min | 1000.0 |
| default seek, mm/min | 3000.0 |
| step port invert mask, int:11000000 | 192.0 |
| step idle delay, msec | 25.0 |
| acceleration, mm/sec^2 | 500.0 |
| junction deviation, mm | 0.05 |
| arc, mm/segment | 0.1 |
| n-arc correction, int | 25.0 |
| n-decimals, int | 3.0 |
| report inches, bool | 0.0 |
| auto start, bool | 1.0 |
| invert step enable, bool | 0.0 |
| hard limits, bool | 0.0 |
| homing cycle, bool | 0.0 |
| homing dir invert mask, int:0000000 | 0.0 |
| homing feed, mm/min | 500.0 |
| homing seek, mm/min | 500.0 |
| homing debounce, msec | 100.0 |

Figura 26 - Machine Settings.

Fonte: Elaborado pelo autor.

- **Control** (Figura 27): refere-se ao controle de movimentos dos eixos X, Y e Z da fresadora CNC. A aba *Step* é disponibilizada para que o usuário possa alterar a quantidade de pulsos a serem enviados para o Arduino Mega2560. Os botões *Zero Position* e *Go Home* são responsáveis por zerar a posição da fresadora, assim como orientar a máquina ao seu ponto inicial para que possa dar início ao processo de usinagem. O ícone *Spindle* tem como em sua função habilitar e desabilitar o pulso de +5V no Arduino Mega2560.



Figura 27 - Control.

Fonte: Elaborado pelo autor.

- **Console** (Figura 28): mediante o Console o usuário pode tanto visualizar cada linha de chamada que está sendo executada como também inserir comando manualmente na fresadora CNC.

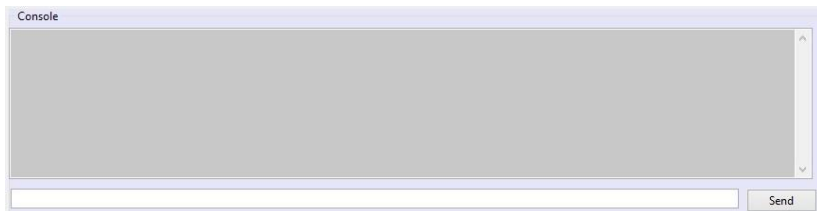


Figura 28 - Console.

Fonte: Elaborado pelo autor.

4.3.3. Programação da máquina aderente ao padrão STEP-NC

Para desenvolver a programação juntamente com o interpretador do arquivo físico STEP-NC de leitura e envio das trajetórias de ferramenta para a máquina, foi utilizado o *software* da Atmel Studio 6, que é uma plataforma de desenvolvimento integrado (IDP) para o desenvolvimento em microcontroladores Atmel AVR. Ele possui um ambiente simples facilitando o uso para escrever, compilar e depurar programas escritos em linguagens C/C++ ou Assembly. É um *software* de licença gratuita e é integrado com o *software* Framework Atmel (ASF), o qual disponibiliza uma grande biblioteca em seu código fonte. O Atmel Studio 6 é o *software* responsável pela compilação de programas na linguagem C/C++ e gravação do arquivo “.hex” no microcontrolador Atmega2560 do Arduino Mega2560.

A programação do padrão STEP-NC foi baseada nas normas (ISO 14649-11) e (ISO 14649-111), juntamente com o programa já desenvolvido denominado GRBL. Basicamente foram implementados novos arquivos no formato “.c” e bibliotecas “.h” para que o GRBL se tornasse aderente ao padrão STEP-NC. Dos arquivos existentes, foram retirados os arquivos *gcode.c* e *gcode.h* e adicionados os arquivos *parametrosgerais.h*, *step_nc.c*, *step_nc.h*, *step_nc_tools.c*, *step_nc_tools.h* e *stepnc_ini.h*, os quais são descritos abaixo:

- *parametrosgerais.h*: Esta biblioteca foi criada para informar ao compilador que ele está executando o MAKEFILE a uma velocidade de (#define F_CPU 16000000L /16 Mhz), para que o AVRDUDE grave o arquivo (.hex) no microcontrolador Atmega2560 conforme APÊNDICE B.
- *step_nc.c* (Figura 29): É o arquivo onde foi programada a leitura e interpretação das entidades do padrão STEP-NC contidas no arquivo físico no formato p21, assim como as transformações de trajetórias nos eixos X, Y e Z contidas no arquivo físico p21 gerado pelo *software* STEP Modeler em sinais de pulso e direção.



```

step_nc.c $
//lista de comandos STEP-NC
char* STEP_NC_COMMANDS[] ={
"PROJECT",
"WORKPLAN",
"MACHINING_WORKINGSTEP",
"CENTER_DRILLING",
"CARTESIAN_POINT",
"MILLING_TECHNOLOGY",
"ROUND_HOLE",
"AXIS2_PLACEMENT_3D",
"MILLING_MACHINE_FUNCTIONS",
"DRILLING",
"BOTTOM_AND_SIDE_ROUGH_MILLING"};
step_nc_pilha pile_cmd;
char currentCommand[50];

static void step_nc_clearCommand(){
/*deleta o ultimo comando executado para poder executar novos comandos*/

for(int i=0;i<50;i++){
currentCommand[i]=0;
}
}
static step_nc_comando* getLastCommand(){
/* retorna o ultimo comando STEP-NC da pilha para ser utilizado */

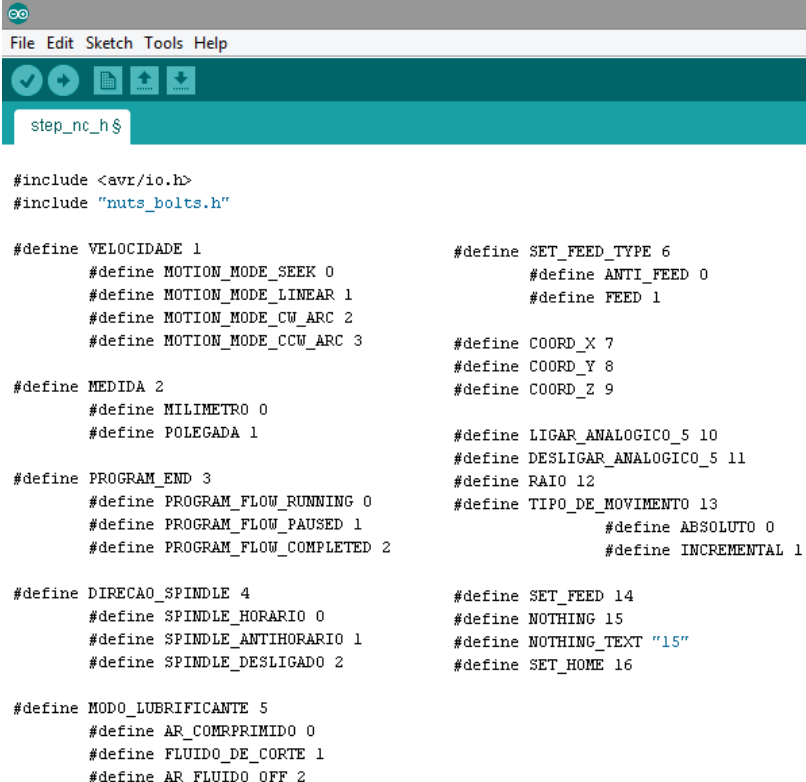
return &pile_cmd.elemento[pile_cmd.numero_elementos-1];
}

```

Figura 29 - Comandos implementados para interpretação do padrão STEP-NC no arquivo step_nc.c.

Fonte: Elaborado pelo autor.

- step_nc_tools.c (Figura 31): É a implementação da estrutura de dados (pilha).



```

#include <avr/io.h>
#include "nuts_bolts.h"

#define VELOCIDADE 1
    #define MOTION_MODE_SEEK 0
    #define MOTION_MODE_LINEAR 1
    #define MOTION_MODE_CW_ARC 2
    #define MOTION_MODE_CCW_ARC 3

#define MEDIDA 2
    #define MILIMETRO 0
    #define POLEGADA 1

#define PROGRAM_END 3
    #define PROGRAM_FLOW_RUNNING 0
    #define PROGRAM_FLOW_PAUSED 1
    #define PROGRAM_FLOW_COMPLETED 2

#define DIRECAO_SPINDLE 4
    #define SPINDLE_HORARIO 0
    #define SPINDLE_ANTIHORARIO 1
    #define SPINDLE_DESLIGADO 2

#define MODO_LUBRIFICANTE 5
    #define AR_COMPRIMIDO 0
    #define FLUIDO_DE_CORTE 1
    #define AR_FLUIDO_OFF 2

#define SET_FEED_TYPE 6
    #define ANTI_FEED 0
    #define FEED 1

#define COORD_X 7
#define COORD_Y 8
#define COORD_Z 9

#define LIGAR_ANALOGICO_5 10
#define DESLIGAR_ANALOGICO_5 11
#define RATIO 12
#define TIPO_DE_MOVIMENTO 13
    #define ABSOLUTO 0
    #define INCREMENTAL 1


#define SET_FEED 14
#define NOTHING 15
#define NOTHING_TEXT "15"
#define SET_HOME 16

```

Figura 30 - Definição de parâmetros na biblioteca *step_nc.h*.

Fonte: Elaborado pelo autor.

- [step_nc_tools.h](#) (Figura 32): Biblioteca utilizada para a estrutura de dados (pilha).



```

void adicionarPilha(step_nc_pilha *pilha ,step_nc_comando elemento)
    if(pilha->numero_elementos>=MAXSIZE){
        printString("ERRO Passou do limite de pilha ");
    }else{
        pilha->elemento[pilha->numero_elementos]=elemento;
        pilha->numero_elementos++;
    }
}

void adicionarFila(step_nc_fila *fila, step_nc_comando elemento){
    if(fila->numero_elementos>=MAXSIZE){
        printString("ERRO Passou do limite de fila");
    }else{
        fila->elemento[fila->numero_elementos]=elemento;
        fila->numero_elementos++;
    }
}

step_nc_comando retirarPilha(step_nc_pilha *pilha ){
    if(pilha->numero_elementos<=0){
        printString("Erro, pilha esta sem elementos");
    }else{
        pilha->numero_elementos--;
    }
    return pilha->elemento[pilha->numero_elementos];
}

step_nc_comando retirarFila(step_nc_fila *fila){
    step_nc_comando temp= fila->elemento[0];
    if(fila->numero_elementos<=0){
        printString("ERRO fila esta sem elementos");
    }
}

```

Figura 31 - Programação da pilha e da fila no arquivo *step_nc_tools.c*.

Fonte: Elaborado pelo autor.

- step_nc_tools.h (Figura 32): Biblioteca utilizada para a estrutura de dados (pilha).



```
step_nc_tools.h

float var[6];
char parametros[PARAMETERSIZE];
}step_nc_comando;

typedef struct{
    uint8_t numero_elementos;
    step_nc_comando elemento[MAXSIZE];
}step_nc_fila;

typedef struct{
    uint8_t numero_elementos;
    step_nc_comando elemento[MAXSIZE] ;
}step_nc_pilha;

void step_nc_tools_init();
void adicionarPilha(step_nc_pilha *pilha ,step_nc_comando elemento);

void adicionarFila(step_nc_fila *fila, step_nc_comando elemento);

step_nc_comando retirarPilha(step_nc_pilha *pilha );


step_nc_comando retirarFila(step_nc_fila *fila);

#endif
```

Figura 32 - Programação da pilha e da fila no arquivo (*step_nc_tools.h*).

Fonte: Elaborado pelo autor.

- stepnc_ini.h (Figura 33): Tem a função de Ligar/Desligar 7 dos 8 relés da placa seletora de sinais.



```

stepnc_ini.h $
#include "settings.h"

#ifndef STEPNC_INI
#define STEPNC_INI
//Ligar/Desligar relays
void step_enable_port_init()
{
    STEP_NC_ACTIVATE_DDR |= (1<<STEP_NC_ACTIVATE_BIT);
    STEP_NC_ACTIVATE_PORT &= ~(1<<STEP_NC_ACTIVATE_BIT);
}

void step_disable_port_init()
{
    STEP_NC_ACTIVATE_PORT |= (1<<STEP_NC_ACTIVATE_BIT);
}

#endif

```

Figura 33 - Programação realizada para habilitar 7 dos 8 relés da placa seletora de sinais.

Fonte: Elaborado pelo autor.

Ao ser definido o desenho da peça, assim como a estratégia de usinagem no *software* CAD (STEP Modeler), o mesmo gera o código no formato físico STEP parte 21 (ou “.p21”) usando a norma (ISO 14649-11). Esse arquivo passa por um módulo de interpretação, que executa e envia as trajetórias para os três eixos da fresadora CNC.

A programação foi realizada a partir de cada solicitação dos comandos contidos no arquivo *STEP-NC.txt*. O *software* STEP-NC Controller envia o arquivo no formato STEP-NC para o Arduino Mega2560, que inicialmente verifica se o arquivo enviado contém comandos do padrão STEP-NC. Se o arquivo contiver um comando inválido, ele envia uma mensagem de comando inválido para o console

do STEP-NC Controller para que o usuário esteja ciente de que o arquivo não foi lido e que não será executado. Se forem identificados comandos STEP-NC no arquivo, inicia-se o projeto e o código é adicionado a uma pilha, e os comandos são sempre executados a partir do topo da pilha.

Os comandos contidos no arquivo STEP-NC solicitam outros novos comandos, adicionando-se assim a próxima linha de comando no topo da pilha. Quando o comando for totalmente executado ele é retirado da pilha, passando-se ao próximo comando. A cada comando identificado, o Arduino Mega2560 envia comandos para o interpretador contido no programa alterado GRBL (mais precisamente no arquivo *stepper.c*), que transforma as trajetórias das coordenadas X, Y e Z em sinais de pulso e direção para cada um dos três eixos da fresadora CNC.

Cada operação a realizar sobre uma ou mais *features* é definida uma instrução chamada *workingstep*. Estes *workingsteps* fornecem as bases para o funcionamento dos planos de trabalho definidos em instruções chamadas *workplan* que definem a fabricação do componente (BENAVENTE *et al.*, 2013).

Como a entidade PROJECT é sempre a primeira entidade a ser lida e executada no Arduino Mega2560, é por ele que são chamadas as outras entidades, funções e comandos, os quais foram adicionados na programação no código do arquivo *step_nc.c* para a movimentação da fresadora. Dentre essas entidades tem-se:

```
# PROJECT
# WORKPLAN
# MACHINING_WORKINGSTEP
# CENTER_DRILLING
# CARTESIAN_POINT
# MILLING_TECHNOLOGY
# ROUND_HOLE
# AXIS2_PLACEMENT_3D
# MILLING_MACHINE_FUNCTIONS
# DRILLING
# BOTTOM_AND_SIDE_ROUGH_MILLING
```

As chamadas de cada uma das entidades que estão contidas dentro do arquivo físico no formato p21 foram programadas seguindo as orientações estabelecidas pelas normas (ISO 14649-10), (ISO 14649-11) e (ISO 14649-111). Cada entidade possui funções/comandos contidos dentro de um parâmetro, o qual pode conter outras novas chamadas de

novas entidades como novas funções/comandos a serem executados na máquina.

Todas as decisões de trajetórias não definidas estão contidas no Arduino Mega2560, pois é nele que está implementado a programação responsável pela leitura das entidades que são responsáveis pelos movimentos da fresadora CNC. O arquivo físico no formato p21 gerado pelo *software* STEP Modeler contém todas as informações para a realização do processo de usinagem. Portanto o Arduino lê apenas as entidades que foram programadas, assim como letras e números que são responsáveis por parâmetros e movimentos que serão realizados na fresadora CNC.

Por exemplo, para se chegar à posição (X115.0, Y80.0) conforme a Figura 34, foi preciso percorrer as chamadas de entidades e suas respectivas funções/comandos contidos em cada parâmetro (PROJECT, WORKPLAN 'Main workplan', WORKPLAN 'Workplan setup', MACHINING_WORKINGSTEP, ROUND_HOLE, AXIS2_PLACEMENT_3D e CARTESIAN_POINT), já o cifrão \$ não é lido pelo Arduino Mega2560.

```
#1=PROJECT('novo projeto',#78,(#4),#2,$,$);
#78=WORKPLAN('Main workplan',(#77),$,$,$);
#77=WORKPLAN('workplan setup
      XY',(#50,#61,#76),$,#12,$);
#76=MACHINING_WORKINGSTEP('FresamentoCircular_R
      GH',#17,#37,#62,$);
#37=ROUND_HOLE('Furo com base
      plana',#4,(#27,#51,#62),#38,#48,#42,$,#49);
#38=AXIS2_PLACEMENT_3D('Furo com base plana
      placement',#39,#40,#41);
#39=CARTESIAN_POINT("(,115.0,80.0,30.0));
```

```

Fresamento Circular (STEP-NC) - Sequência.txt - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
#1=PROJECT('novo projeto',#78,(#4),#2,$,$);
  #78=WORKPLAN('Main workplan',(#77),$,$,$);
    #77=WORKPLAN('workplan setup XY',(#50,#61,#76),$,$,#12,$);
      #50=MACHINING_WORKINGSTEP('FuroCentro_RGH',#17,#37,#27,$);
        #17=PLANE('security plane',#13);
          #13=AXIS2_PLACEMENT_3D('',#14,#15,#16);
            #14=CARTESIAN_POINT('',(0.0,0.0,55.0));
            #15=DIRECTION('',(0.0,0.0,1.0));
            #16=DIRECTION('',(1.0,0.0,0.0));
          #37=ROUND_HOLE('Furo com base plana',#4,(#27,#51,#62),#38,#48,#42,$,#49);
            #4=WORKPIECE('workpiece',#9,50.0,$,$,#11,());
            #9=MATERIAL('SAE 1020','ACO SEM LIGA',(#10));
            #10=NUMERIC_PARAMETER('Hardness',250.0,'HB');
            #11=BLOCK('piece',#5,200.0,150.0,30.0);
            #5=AXIS2_PLACEMENT_3D('workpiece placement',#6,#7,#8);
              #6=CARTESIAN_POINT('',(0.0,0.0,0.0));
              #7=DIRECTION('',(0.0,0.0,1.0));
              #8=DIRECTION('',(1.0,0.0,0.0));
            #27=CENTER_DRILLING($,$,'Center Drilling',5.0,#28,#29,#34,#35,$,3.0,$,$,#36);
              #28=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
              #29=MILLING_CUTTING_TOOL('center drill',#30,(#32),$,$,$);
                #30=CENTER_DRILL(#31,2,-RIGHT,$,25.0);
                #31=TOOL_DIMENSION(2.0,0.5235987755982988,0.0,1.0,0.0,0.0,0.0);
                #32=CUTTING_COMPONENT(50.0,#33,$,$,$);
                #33=MATERIAL('P',$,$,$);
                #34=MILLING_TECHNOLOGY(90.0,TCP,.110,0.7200,0,$,.F.,.F.,.F.,$);
                #35=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,$.T.,$,$,$);
                #36=DRILLING_TYPE_STRATEGY($,$,$,$,$);
              #51=DRILLING($,$,'Drilling',5.0,#59,#52,#57,#58,$,10.0,$,$,#60);
                #59=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
                #52=MILLING_CUTTING_TOOL('twist drill',#53,(#55),$,$,$,$);
                  #53=TWIST_DRILL(#54,2,-RIGHT,$,60.0);
                  #54=TOOL_DIMENSION(1.5,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
                  #55=CUTTING_COMPONENT(70.0,#56,$,$,$,$);
                  #56=MATERIAL('P',$,$,$);
                #57=MILLING_TECHNOLOGY(115.0,TCP,.110,0.7350,0,$,.F.,.F.,.F.,$);
                #58=MILLING_MACHINE_FUNCTIONS(.F.,$,$,.F.,$,$.T.,$,$,$);

```

Figura 34 - Fragmento da sequência de programação do padrão STEP-NC para o fresamento circular.

Fonte: Elaborado pelo autor.

4.4. OPERAÇÃO REMOTA DA MÁQUINA VIA INTERNET

O Grupo de Integração da Manufatura (GRIMA) possui um servidor no qual está hospedado o *software* STEP Modeler, responsável pela modelagem da peça CAD e pela determinação de parâmetros de corte, ferramentas de corte a serem utilizadas, bem como a geração do código STEP-NC. Porém, para proporcionar que o usuário possa operar remotamente a fresadora CNC via Internet, foi desenvolvido um programa na linguagem C utilizando-se uma biblioteca da linguagem Java (aproveitando a vantagem de que programas em Java podem ser executados em qualquer sistema operacional).

Com isso, foi desenvolvido um servidor local (STEP-NC *Server*) versão 1.7 (Figura 35), que começa a executar junto com o sistema operacional Windows, já estabelecendo uma ponte entre o cliente e o Arduino Mega2560 no computador instalado no painel de controle da fresadora CNC. No ícone *Port* é selecionada a porta do *socket* utilizado na conexão de transmissão de dados do STEP-NC.

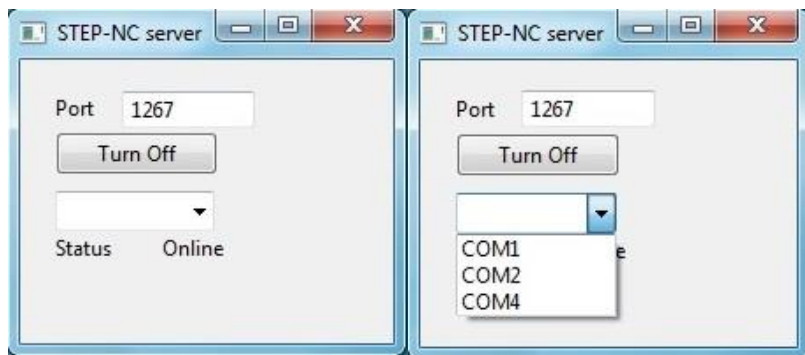
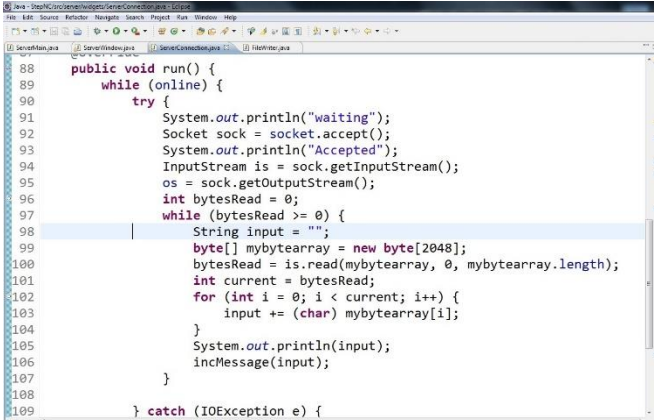


Figura 35 - STEP-NC server.

Fonte: Elaborado pelo autor.

O servidor disponibiliza uma caixa de seleção na qual o usuário local pode acessar a porta COM utilizada para realizar também a transmissão local do código STEP-NC via porta serial para o Arduino. Além desta caixa de seleção, o servidor disponibiliza outros dois campos nos quais o usuário insere o IP local da máquina onde está o Arduino junto com a porta de rede.

Essa ponte libera acesso para que o usuário tenha total controle via interface do STEP-NC Controller. Todas as mensagens do cliente ao servidor são redirecionadas via porta serial para o Arduino Mega2560 e vice-versa. Foram utilizadas as bibliotecas *Java.io (InputStream e OutputStream)* para recebimento de dados e a transmissão via *sockets, Java.net.Socket*, usando o protocolo TCP (*Transmission Control Protocol*), conforme mostrado na Figura 36. O servidor possui uma função que armazena a última configuração de portas utilizadas para que sempre haja conexão entre o cliente e o Arduino Mega2560.

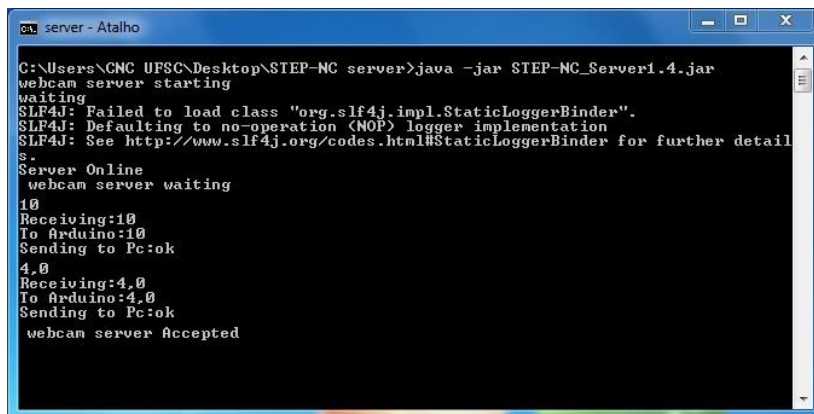


```
88     public void run() {
89         while (online) {
90             try {
91                 System.out.println("waiting");
92                 Socket sock = socket.accept();
93                 System.out.println("Accepted");
94                 InputStream is = sock.getInputStream();
95                 os = sock.getOutputStream();
96                 int bytesRead = 0;
97                 while (bytesRead >= 0) {
98                     String input = "";
99                     byte[] mybytearray = new byte[2048];
100                    bytesRead = is.read(mybytearray, 0, mybytearray.length);
101                    int current = bytesRead;
102                    for (int i = 0; i < current; i++) {
103                        input += (char) mybytearray[i];
104                    }
105                    System.out.println(input);
106                    incMessage(input);
107                }
108            } catch (IOException e) {
```

Figura 36 - Fragmento da programação de conexão via rede do STEP-NC server.

Fonte: Elaborado pelo autor.

Todas as vezes que o usuário realizar a comunicação remota com o servidor (STEP-NC *server*), uma mensagem irá aparecer conforme mostrado na Figura 37.



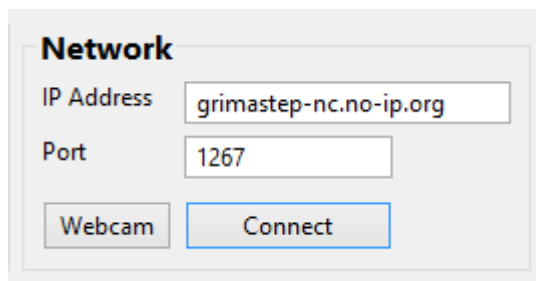
```
C:\Users\CNC UFSC\Desktop\STEP-NC server>java -jar STEP-NC_Server1.4.jar
webcam server starting
waiting
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Server Online
webcam server waiting
10
Receiving:10
To Arduino:10
Sending to Pc:ok
4.0
Receiving:4.0
To Arduino:4.0
Sending to Pc:ok
webcam server Accepted
```

Figura 37 - Mensagem na tela do servidor no arquivo “.bat”.

Fonte: Elaborado pelo autor.

Para que o usuário possa realizar a comunicação remota com o servidor, foram disponibilizadas duas opções de conexão:

1. Manual: Digitando o atual IP (150.162.27.114) da fresadora na UFSC no campo IP Address (Figura 38), sempre utilizando a *Port* 1267.



Network

IP Address

Port

Figura 38 - Conexão manual com o STEP-NC *server*.

Fonte: Elaborado pelo autor.

1. Automático: Clicando no botão *Connect*, onde não há a necessidade de digitar-se manualmente o IP (150.162.27.114) do computador da fresadora CNC. Esta função foi adicionada porque cada vez que o IP do servidor central da UFSC é alterado, todos os IPs são renovados, fazendo com que o usuário tenha que realizar uma busca do IP na máquina local para poder ter acesso ao servidor STEP-NC *server* da fresadora.

Para que a conexão automática pudesse ser realizada foi desenvolvido um registro no site No-IP (www.noip.com) com o nome: grimastep-nc.no-ip.org, que oferece uma maneira rápida e de fácil configuração para a execução de qualquer tipo de servidor para a Web. O No-IP contém um programa DUC (*Dynamic DNS Update Client*) (Figura 39), que é instalado no computador no qual é iniciado juntamente com o sistema operacional Windows 7. Ele notifica toda vez que o endereço IP do computador é alterado, fazendo com que o usuário não precise saber qual o IP do computador da fresadora CNC, facilitando assim a conexão do *software* STEP-NC Controller com o servidor (STEP-NC *server*).



Figura 39 - Programa DUC para a conexão entre o software STEP-NC Controller e o servidor STEP-NC *server*.

Fonte: Elaborado pelo autor.

Para a captura de imagem na câmera webcam (Figura 40), foi utilizada como parte da programação a biblioteca *com.github.sarxos.webcam.Webcam*, adicionada ao *software* Eclipse para se conectar ao servidor STEP-NC server.

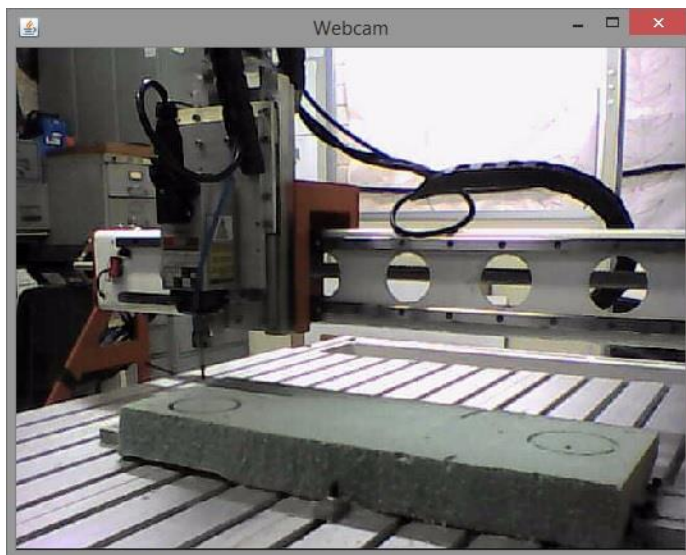


Figura 40 - Imagem feita pela webcam da fresadora CNC.

Fonte: Elaborado pelo autor.

No STEP-NC Controller, a janela da *webcam* na parte do cliente apresenta uma pequena pausa depois de um certo tempo aleatório, ela ocorre devido ao fato de ter sido utilizado a transmissão da imagem por TCP e uma outra forma de transmissão é a por UDP (*User Datagram Protocol*) que não foi implementado devido à falta de suporte adequado para inserção desta biblioteca de *webcam* no ambiente Java.

Não foi criado neste trabalho um *streaming* de vídeo entre o servidor e cliente. Assim, o servidor captura fotos tiradas pela *webcam*, converte-as em *bufferedimage* e envia-as para o cliente usando *sockets* na porta escolhida +1 (só pode-se transmitir um tipo de dados por *sockets*). As imagens são enviadas por *sockets* usando-se TCP (*Transmission Control Protocol*), que resulta em um atraso significativo dependendo do resultado do comando *ping*, o qual é usado no âmbito do TCP/IP para solucionar problemas de conectividade, acessibilidade e resolução de nomes.

A Figura 41 contém o fluxograma do sistema implementado para o padrão STEP-NC na fresadora CNC e seus respectivos módulos de processamento.

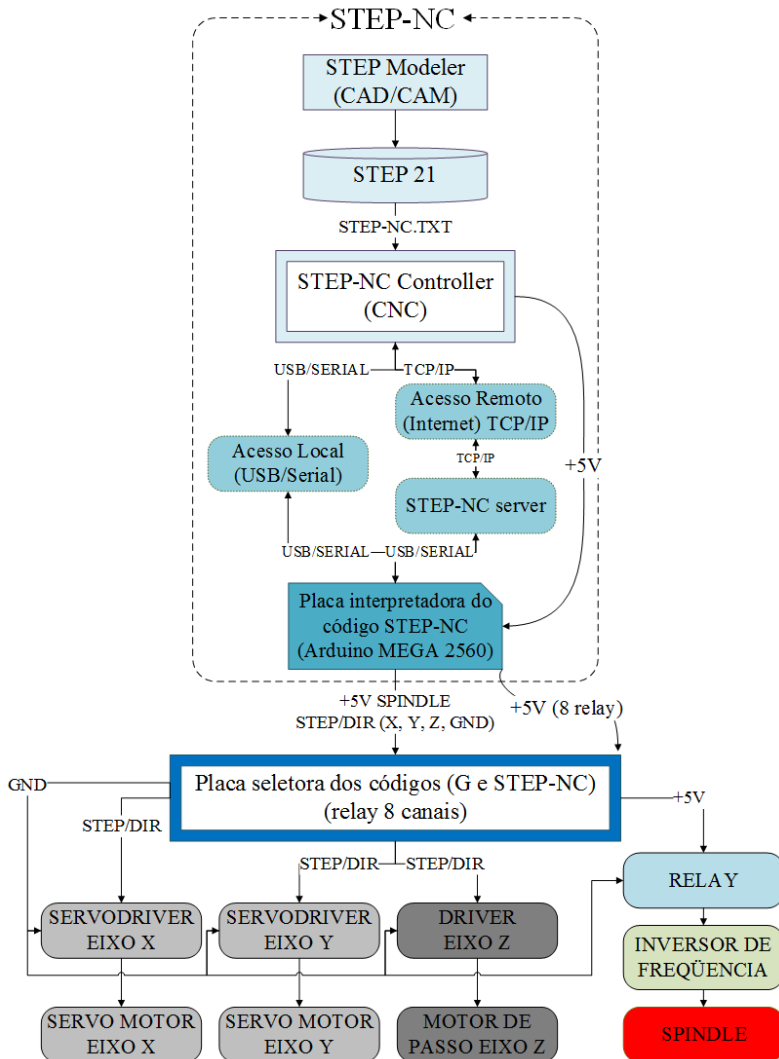


Figura 41 - Fluxograma geral do padrão STEP-NC implementado na fresadora CNC.

Fonte: Elaborado pelo autor.

A Figura 42 contém o fluxograma geral do sistema implementado, da integração dos dois padrões (código G e STEP-NC) na fresadora CNC e seus respectivos módulos de processamento.

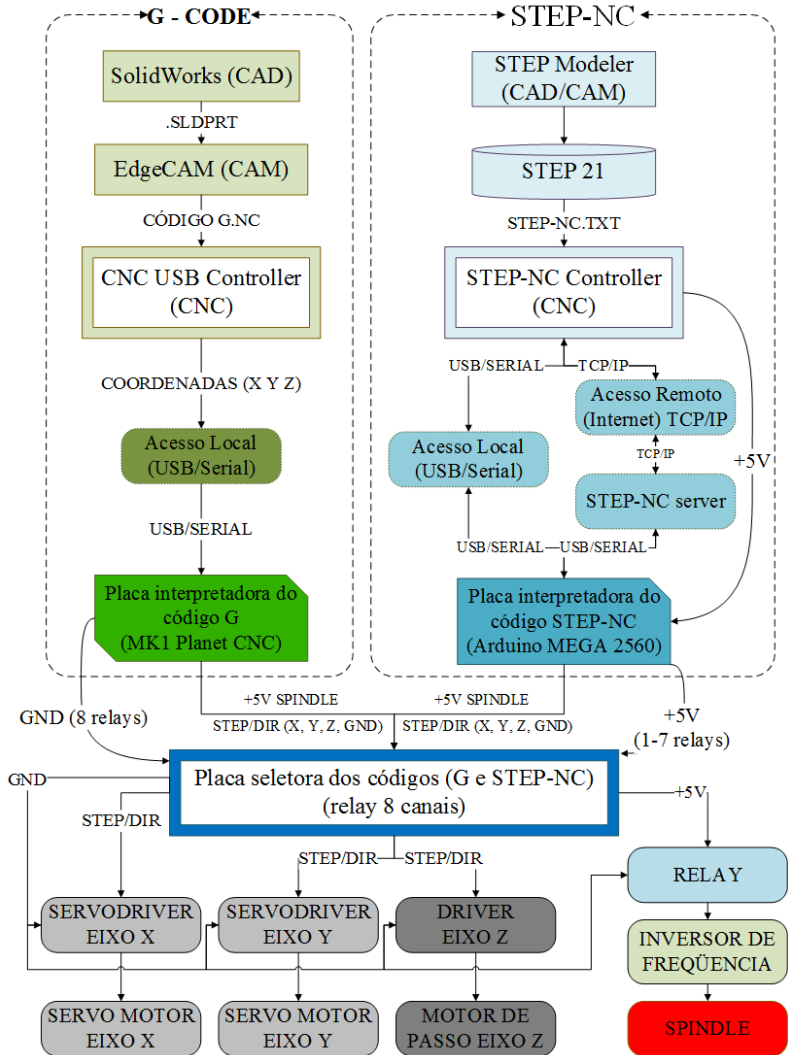


Figura 42 - Fluxograma geral dos padrões implementados.

Fonte: Elaborado pelo autor.

5. VALIDAÇÃO EXPERIMENTAL POR USINAGEM

Para a análise completa de toda linguagem de baixo nível da programação em C que foi utilizado no arquivo GRBL, assim como a placa seletora de sinais, foram executadas as *features* da Tabela 2 para os testes experimentais.

Dos arquivos gerados pelo *software* STEP Modeler no formato STEP-NC p21, somente as *features* da Tabela 2 foram avaliadas, levando em consideração (tempo, acabamento e dimensões da peça). Não sendo avaliados problemas decorrentes da usinabilidade de materiais nem aspectos de precisão dimensional.

Tabela 2 – Features funcionais do STEP-NC Modeler funcionais na fresadora CNC.

| Features | | Funcional | | Implementado |
|---------------------|------------------|-----------|-----|-------------------------|
| | | SIM | NÃO | |
| FURO | Base Plana | X | | Sim |
| | Base Cônica | X | | Sim |
| | Base Esférica | X | | Sim |
| | Base arredondada | X | | Sim |
| | Cônico | X | | Sim |
| RANHURA | Retangular | X | | Sim |
| | Circular | X | | Com pausas no movimento |
| CAVIDADE RETANGULAR | Fundo plano | X | | Com pausas no movimento |

Foram realizados testes de usinagem dos processos de furação, ranhura retangular e fresamento circula. Testes estes com a finalidade de validar o funcionamento do *software* STEP-NC Controller utilizando uma peça contendo *features* disponibilizadas pelo *software* STEP Modeler (mediante o arquivo físico p21).

Durante o processo de usinagem de ranhura retangular e furação, conforme mostrado na Figura 43 e Figura 44, foi utilizado o código STEP-NC do APÊNDICE C e o código G do APÊNDICE D. Em ambos os processos, a fresadora CNC não apresentou diferença no tempo de usinagem e acabamento.

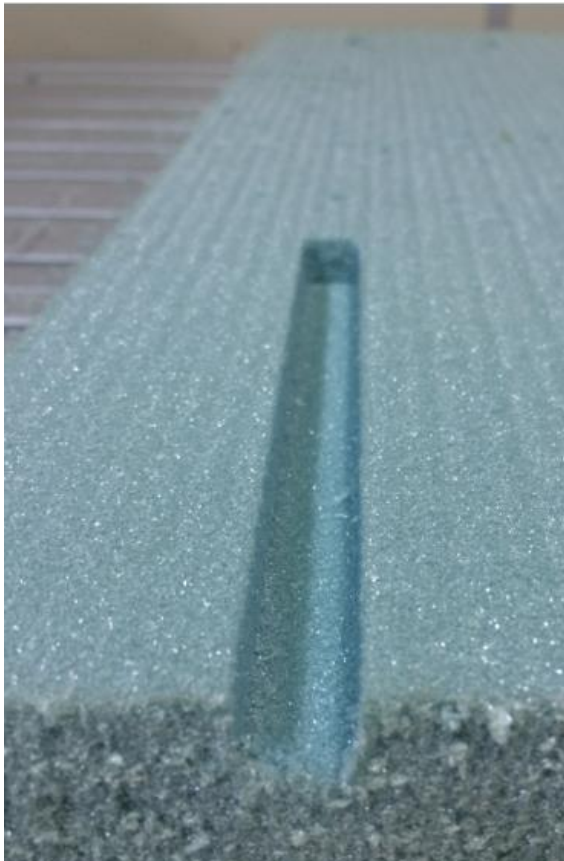


Figura 43 - Usinagem em espuma de PVC do padrão STEP-NC no processo de ranhura retangular.

Fonte: Elaborado pelo autor.



Figura 44 - Usinagem em espuma de PVC do padrão STEP-NC no processo de furação.

Fonte: Elaborado pelo autor.

No caso do fresamento circular, a fresadora CNC apresentou pequenas pausas durante a execução dos movimentos de coordenadas (X,Y e Z). Foram utilizadas para a usinagem duas *features* de usinagem (furação e fresamento circular) devido a uma limitação do *software* STEP Modeler na função circular. Na Figura 45 consta o fresamento circular no qual foi gerado o arquivo físico p21 mostrado no APÊNDICE E, onde obteve-se o mesmo tempo de usinagem se comparado com o código G do APÊNDICE F.

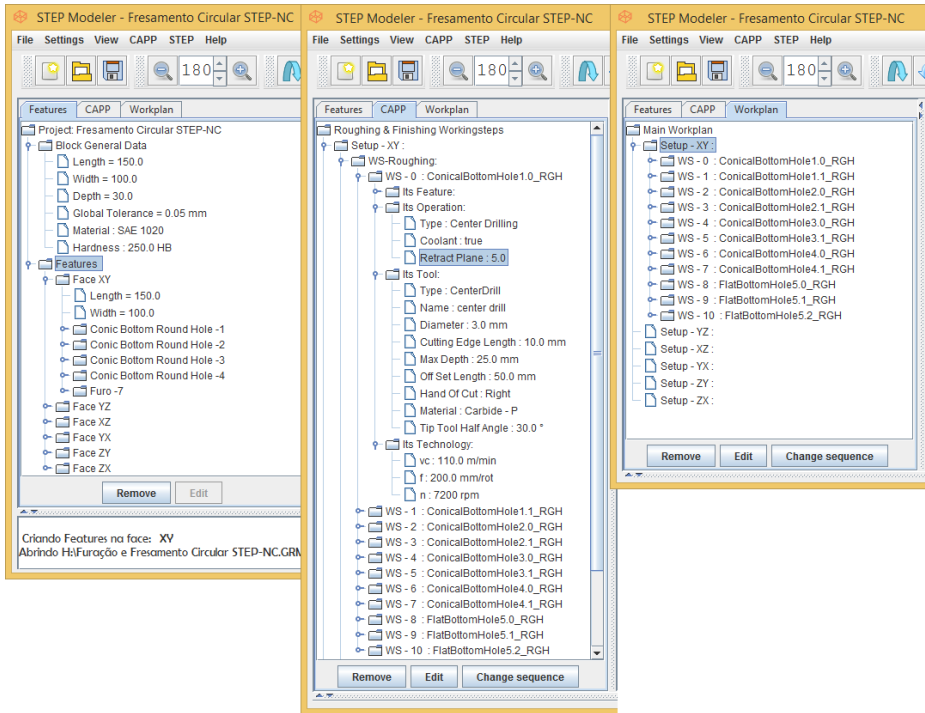


Figura 45 - Features, CAPP e Workplan do processo de fresamento circular.

Fonte: Elaborado pelo autor.

Durante os testes, a fresadora CNC teve um desempenho superior devido à interoperabilidade, integração de CAD/CAM/CNC para confecção da peça até a geração do arquivo p21 no padrão STEP-NC do que em relação ao projeto do mesmo para o código G.

O desempenho inferior se deu com relação ao tempo de usinagem, no código G o tempo total da usinagem foi de 3 minutos enquanto que no STEP-NC o tempo foi de 3 minutos e 30 segundos, uma vez que houve movimentos travados nos eixos X e Y, o que não ocorreu usando-se o código G. Supõe-se que isso pode ter acontecido em função do arquivo *open source*, cuja programação inclui a conversão de caracteres (letras e números) para sinais de pulso e direção, assim como a modulação PWM (*Pulse-Width Modulation*) que é de baixo nível. Como o programa GRBL possui todos os cálculos matemáticos para a movimentação dos eixos dedicados ao código G, supõe-se que as pausas estejam sendo causadas na parte em que foi programado a transformação de caracteres em sinais de pulso e direção, e não em função da programação do STEP-NC que foi implementada. Um vídeo está disponível no Youtube (<http://youtu.be/CBIu0IPfXZw>), o qual mostra o procedimento de usinagem contendo as pausas mencionadas acima.

Isso pode ser explicado porque, no caso da implementação para código G, o Arduino Mega2560 recebe o arquivo inteiro e executa os movimentos a partir da memória que foi definida para este modelo de microcontrolador, enquanto que no STEP-NC, ao invés de ler-se a linha da memória, seleciona-se qualquer linha da memória para o computador onde está o STEP-NC Controller. O motivo de não se enviar todo o arquivo para o Arduino é porque só é permitido executar um movimento para cada linha: enquanto no código G cada linha corresponde a um movimento, no STEP-NC frequentemente é necessária a leitura de duas ou mais linhas para se estabelecer um movimento e, portanto, o envio do arquivo de texto inteiro não é suficiente.

As equações (1) e (2), são utilizadas para gerar as coordenadas X e Y no padrão STEP-NC para que a fresadora CNC possa realizar o processo de fresamento circular.

$$X = \text{raio} * \text{Math.cos}(\text{anguloInicial} + i * \text{dAngulo}); \quad (1)$$

$$Y = \text{raio} * \text{Math.sin}(\text{anguloInicial} + i * \text{dAngulo}); \quad (2)$$

Foi retirado da programação aplicada no *software* Eclipse um exemplo da equação que gera as trajetórias (X e Y) para o fresamento circular:

```
public static Ponto[] determinarPontosEmCircunferencia(double
anguloInicial, double deltaAngulo, double raio, int numeroDePontos,
double zAtual)
```

onde:

- $\hat{\text{Ângulo Inicial}}$ → ângulo onde se inicia a interpolação
- $\text{Delta } \hat{\text{Ângulo}}$ → ângulo percorrido
- Raio → raio de interpolação do arco circular
- Número de pontos → número de pontos de interpolação

```
{
Ponto[] saida = new Ponto[numeroDePontos + 1];
double x, y, dAngulo = 0;

dAngulo = deltaAngulo / numeroDePontos;
for(int i = 0; i < numeroDePontos + 1; i++)
{
x = raio * Math.cos(anguloInicial + i * dAngulo);
y = raio * Math.sin(anguloInicial + i * dAngulo);
saida[i] = new Ponto(x, y, zAtual);
}
return saida;
}
```

6. CONCLUSÃO

6.1. CONSIDERAÇÕES FINAIS

Para uma análise final deste trabalho, que buscou tornar uma fresadora CNC aderente ao padrão STEP-NC (ISO 14649-11), foram levados em consideração os objetivos propostos inicialmente. Pode-se afirmar que os resultados foram satisfatórios, e as ferramentas utilizadas para a adequação do padrão STEP-NC na fresadora CNC atenderam a proposta de pesquisa e implementação deste novo padrão na fresadora CNC.

O controlador Arduino Mega2560 na fresadora CNC executou os arquivos físicos no formato STEP-NC parte 21. O padrão STEP-NC é um novo modelo de dados para a troca de informações entre os sistemas CAD/CAPP/CAM e CNC, proporcionando a integração da manufatura. Espera-se que o padrão STEP-NC venha a substituir o inadequado padrão ISO 6983 (código G), tornando-se a nova linguagem de programação para máquinas CNC.

A implementação do *software* para converter o arquivo no formato STEP-NC para a movimentação dos eixos da máquina foi um desafio, uma vez que a programação foi significativamente complexa e extensa. A implementação do algoritmo foi realizada utilizando-se as linguagens C e Java.

A validação do *software* de conversão e, conseqüentemente, do controlador STEP-NC Controller, foi realizada por meio de um estudo de caso de uma peça teste usinada na fresadora CNC aderente ao padrão STEP-NC. Foi efetuada uma comparação entre o uso da implementação usando-se código G com a abordagem usando-se o padrão STEP-NC, que inclui os *workingsteps* do programa STEP-NC parte 21 gerado pelo *software* STEP Modeler (BENAVENTE, 2011), (BENAVENTE *et al.*, 2013).

O STEP-NC foi implementado sobre um programa *open source* denominado GRBL, que contém arquivos que foram programados especificamente para código G, seguindo a norma (ISO 6983-1). Os arquivos contidos no GRBL, que efetuam todos os cálculos matemáticos e transformações das trajetórias em sinais de pulso e direção para a movimentação dos eixos, funcionaram para os processos de furação e fresamento. Entretanto, o mesmo desempenho não foi obtido para o padrão STEP-NC, principalmente no fresamento circular, que apresenta pausas entre cada segmento de reta devido principalmente à

transformação em sinais de pulso e direção basear-se no código G e não no STEP-NC.

Considera-se que a implementação usando-se o padrão STEP-NC venha a obter um desempenho equivalente ou até melhor do que a implementação baseada no código G se a programação não for baseada nos arquivos responsáveis pela transformação das trajetórias em sinais de pulso e direção contidos no programa GRBL (*open source*). A substituição da linguagem de programação atualmente em C (estruturada) para C++ (orientada a objeto) é um caminho para um *retrofitting* da programação do padrão STEP-NC na fresadora CNC.

As pausas encontradas durante o processo de fresamento circular do padrão STEP-NC ocorrem porque existe uma diferença de chamada das entidades entre os dois códigos (código G e STEP-NC). No código G o programa GRBL recebe o arquivo “.txt” inteiro e executa os movimentos a partir da memória do Arduino, enquanto que no STEP-NC, ao invés de ler-se a linha da memória, seleciona-se e lê-se a linha a partir de uma pilha na qual cada entidade é atribuída a uma fila. No código G cada linha gera um movimento, enquanto que no STEP-NC isso não acontece devido a cada entidade possuir movimentos e parâmetros que são responsáveis pelos movimentos na fresadora CNC.

A programação para a abordagem baseada no código G foi realizada no Arduino, e o Arduino executa no máximo um movimento por linha de código enviado. No caso do código G isto não ocorreu, pois cada linha do código G gera um movimento. Por outro lado, no caso do arquivo STEP-NC nem todas as linhas realizam movimentos, sendo muitas vezes necessárias duas ou mais linhas para a realização de um movimento.

Como solução para as pausas nos movimentos (X,Y e Z) da fresadora CNC, a programação do padrão STEP-NC no Arduino foi refeita buscando eliminar estas pausas encontradas no processo de fresamento circular, que podem ser visualizadas tanto no momento da usinagem assim como executando o arquivo (.bat) do STEP-NC server, arquivo este que permite ao usuário acompanhar todas as chamadas das entidades contidas no arquivo p21 que são responsáveis pelos movimentos da fresadora CNC durante o processo de usinagem.

6.2. SUGESTÕES PARA TRABALHOS FUTUROS

Para trabalhos futuros sugere-se novas pesquisas de maneira a proporcionar que a fresadora CNC se torne completamente aderente ao padrão STEP-NC, tais como:

- Realizar a otimização da programação atual do Arduino Mega2560, adicionando novos algoritmos para otimizar as trajetórias de ferramenta no processo de usinagem no fresamento circular;
- Implementar no painel de controle o microcomputador *raspberry Pi* para uma otimização interna da disposição física dos equipamentos eletrônicos responsáveis pelo funcionamento da fresadora CNC;
- Integrar o STEP-NC Controller juntamente ao sistema STEP Modeler para a completa integração da manufatura entre sistemas CAD/CAPP/CAM e CNC em um único *software*;
- Iniciar uma nova programação na linguagem C++ orientada a objetos do padrão STEP-NC (ISO 14649-11) em um novo microcontrolador que a suporte, substituindo todos os outros arquivos (".c" e ".h") contidos no programa original GRBL;
- Implementar o elemento *Viewer* no *software* STEP-NC Controller com a função de abrir uma janela (GUI de visualização) na qual é exibido o arquivo físico p21 gerado pelo *software* STEP Modeler;
- Implementar o padrão MTConnect na fresadora CNC juntamente com o STEP-NC Controller aderente ao padrão STEP-NC. O MTConnect (www.mtconnect.org) é um conjunto de padrões abertos que visa proporcionar a interoperabilidade entre os controladores de máquinas-ferramenta, dispositivos e aplicações de *software* por meio da publicação de dados sobre redes que utilizam IP (Internet Protocol). Desta forma, dados do chão de fábrica podem ser publicados on-line e monitorados em um lugar remoto (KADIR, XU e RIDWAN, 2011). O padrão MTConnect busca proporcionar o acesso aberto a controladores CNC com uma definição neutra de conteúdos de informação e vias de acesso, e assim o acesso e a comunicação de dados oferecem suporte à interoperabilidade (CAMPOS e MIGUEZ, 2011).

7. REFERÊNCIAS

ADAMCZYK, Z.; KOCIOLEK, K. CAD/CAM Technological Environment creation as an interactive application on the web. **Jounal of materials processing technology**, v. 109, p. 222-228, 2001.

ALLEN, R. D.; NEWMAN, S. T.; ROSSO JR, R. S. U. CAD/CAM solution for STEP-compliant CNC manufacturing., v. 16, n. 7-8, p. 590-597, 2003.

ALVARES, A. J. **Uma metodologia para Integração CAD/CAPP/CAM voltada para manufatura**. Florianópolis. 2003.

ANSI 1978. **IGES - Initial Graphics Exchange Specification (IGES) Layered Electrical Product Application Protocol (AP)**. 1980.

ARDUINO. **Arduino.cc**. Disponível em: <<http://www.arduino.cc/>>. Acesso em: 22 agosto 2014.

BENAVENTE, J. C. T. **Um Sistema Para O Projeto E Fabricação De Peças Mecânicas A Distância Via Internet Aderente À Norma ISO 14649 (STEP-NC)**". UFSC. Florianópolis, p. 129. 2011. (Trabalho de Doutorado em Engenharia Mecânica, Programa de Pós-Graduação em Engenharia Mecânica.).

BENAVENTE, J. C. T. et al. A STEP-NC Compliant System for the Remote Design and Manufacture of Mechanical Components Through the Internet. **International Journal of Computer Integrated Manufacturing**., Florianópolis, Brasil, v. 26, n. 5, p. 412 - 428, Setembro 2013.

BIZELLO, S. A. **Curso AutoCAD 2002**. Autodesk. São Paulo, p. 861. 2001.

BROGUEIRA, J. G.; LOPES, M.; KLEIN, L. Projetos em 2D dão lugar à comunicação visual. **CADesign**, v. 100, n. 10, p. 16-17, Março 2004.

BROWN , S. M.; WRIGHT, P. K. A Progress Report on the Manufacturing Analysis Service, an Internet-Based Reference Tool. **Jounal of Manufacturing Systems**, v. 17, p. 389-401, 1998.

BROWNE, J.; MCMAHON, C. **CAD/CAM, Principles, Practice and Manufacturing Management**. 2ª edição. ed. Wesley: Addison Wesley Longman, 1998.

CALABRESE, F.; CELENTANO, G. Design and Realization of a STEP-NC Compliant CNC Embedded Controller., Itália, Dezembro 2009.

CAMPOS, J. G.; MIGUEZ, L. R. Standard Process Monitoring and Traceability Programming in Collaborative CAD/CAM/CNC Manufacturing Scenarios, *Computers in Industry*. [S.l.]: [s.n.], v. 62, 2011. p. 311-322.

CHANG, T. C.; WANG, H. P.; WYSK, A. **Computer-Aided Manufacturing**. 3. ed. [S.l.]: Prentice Hall International Series in Industrial and Systems Engineering, 2005.

CHEN, X.; HOFFMANN, M. C. **On editability of feature-based design. Computer-Aided Design**. [S.l.]: [s.n.], v. 27, 1995. 900-915 p.

CHOI, et al. STEP-compliant CNC system for turning: data model, architecture, and implementation. **Computer-Aided Design**, 38, 2006. 677-688.

CNC,. <http://www.planet-cnc.com/index.php?page=hardware>. **Planet CNC**. Acesso em: 2013.

DIAS, A.; MATOS DA CUNHA, R. R. **Estudo e desenvolvimento de metodologias na troca de dados CAD/CAM**. Florianópolis: UFSC, 2000.

EASTMAN, C. M. **Out of STEP. Computer-Aided Design**. 5 Edição. ed. [S.l.]: [s.n.], v. 26, 1994. 338-340 p. Julho.

FERREIRA , J. C. E. **Apostila Sistemas Integrados de Manufatura da disciplina de Integração da Manufatura**. Florianópolis: Curso de Pós-Graduação em Engenharia Mecânica, 2005.

FIRMAN, R.; XUN, X.; GUANGYU, L. A framework for machining optimisation based on STEP-NC. **Journal of Intelligent Manufacturing**, Nova Zelândia, v. 23, p. 423-441, 2012 Janeiro.

GAO, ; GINDY, ; ZHENG,. Mathematical representation of features conversion for CAD / CAM systems integration. **Robotics and Computer-Integrated Manufacturing**, 20, October 2004. 457-467.

GROOVER, M. P. *Automation, Production Systems, and Computer-Integrated Manufacturing*. Third Edition. ed. [S.l.]: Prentice Hall., 2007.

HAN , ; HAN, ; YI. **Manufacturing feature recognition toward integration with process planning**. 31. ed. [S.l.]: IEEE Transactions on Systems, Man and Cybernetics., 2001. 370-380 p.

HARBS,. **Controlador CNC aderente a ISO 14649 com ferramentas de software para construção e integração de uma arquitetura de controle baseada na IEC 61499**. Universidade Estadual de Santa Catarina (UDESC).

Joinville, p. 148. 2012. (Dissertação de Mestrado em Engenharia Elétrica - UDESC).

HARDWICK, M.; LOFFREDO, D. Lessons learned implementing STEP-NC AP-238. **International Journal of Computer Integrated Manufacturing**, USA, p. 523-532, 2009 Dezembro.

HASEGAWA, A. Y. **Controlador numérico de interface aberta STEP-NC**. Universidade do Estado de Santa Catarina. [S.l.]. 2010.

[HTTPS://GITHUB.COM/GRBL/GRBL](https://github.com/GRBL/GRBL). 2012.

ISO 10303-1. **Automation systems and integration -- Product data representation and exchange (STEP - Standard for the Exchange of Product Data)**. 2002.

ISO 10303-11. **Industrial automation systems and integration -- Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual**. 2004.

ISO 10303-21. **Industrial automation systems and integration -- Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure**. 2002.

ISO 14649. **STEP-NC Project, "STEP-Compliant Data Interface for Numerical Controls", ESPRIT Project EP 29708, STEP-NC Newslette., 2000**. <http://www.step-nc.org>.

ISO 14649-1. **STEP-NC Project, "STEP-Compliant Data Interface for Numerical Controls", ESPRIT Project EP 29708, STEP-NC Newslette**. 2003.

ISO 14649-10. **Industrial automation systems and integration -- Physical device control, ISO 14649 Data model for Computerized Numerical Controllers, Part 10: General Process Data**. 2002.

ISO 14649-11. **Industrial automation systems and integration -- Physical device control, ISO 14649 Data model for Computerized Numerical Controllers, Part 11: Process Data For Milling**. 2002.

ISO 14649-111. **Industrial automation systems and integration -- Physical device control, ISO 14649 Data model for Computerized Numerical Controllers, Part 111: Tools For Milling**. 2002.

ISO 14649-TOOLKIT. **Google code, Toolkit for ISO 14649 / STEP-NC.**, 2009. Disponível em: <<http://code.google.com/p/iso-14649-toolkit/>>.

ISO 6983-1. "Automation systems and integration -- Numerical control of machines - Program format and definition of address words - Part 1: Data format for positioning, line motion and contouring control systems. 1982.

KADIR, A. Z. A.; XU, X.; RIDWAN, F. Employing Actual Shop-floor Status and Machine Tool Kinematic Data Model in Machining Simulation. **International Journal of Integrated Engineering**, v. 3, p. 32-37, 2011. ISSN 1. KAO, Y. C.; LIN, G. C. CAD/CAM Collaboration and remote Machining. **Computer Integrated Manufacturing Systems**, v. 9, p. 149-160, 1996.

KEMMERER, J. **STEP The Grand Experience. Manufacturing Engineering Laboratory. National Inst. of Standards and Technology.**, Gaithersburg (MD), 2009.

KOCIOLEK, K. CAD/CAM Tecnological Environment creation as an interactive application on the web. **Jounal of materials processing technology**, v. 109, p. 222-228, 2001.

KRAMER, T. R.; MESSINA, E. R.; PROCTOR, F. M. "The NIST RS274NGC Interpreter - Version 3", 2010.

LEONDES, C. T. **The Design of Manufacturing Systems (Computer-Aided Design, Engineering, and Manufacturing)**. [S.l.]: CRC Press, v. 5, 2002.

LEWIS, J. One Step Closer. **Design News**, 2002. Disponível em: <<http://www.designnews.com/article/CA241152.html>>.

LIANGJI , X. STEP-NC in Support of Machining Process Optimization. **Springer Link**, Seattle, Washington, 2007.

LOFFREDO, D. **Fundamentals of STEP Implementation. STEP Tolls**, Outubro 2000. Disponível em: <<http://www.steptools.com/library/fundimpl.pdf>>. Acesso em: 24 ago. 2014.

MATTHIEU, R. et al. An advancedSTEP-NCcontrollerforintelligentmachiningprocesses. **Elsevier**, França, Korea do Sul, Dezembro 2011.

MAZIERO, N. ; BONATTO, ; PEREZ,. Representação de tolerâncias dimensionais e geométricas num sistema CAD baseado em features rotacionais 2D, Passo Fundo , Abril 2001. 8.

MENEZES, M.; SILVA, E. L. L. **Metodologia da Pesquisa e Elaboração de Dissertação**. 4 edição revisada e atualizada. ed. Florianópolis : UFSC, 2005.

MIAO, H. K.; SHAH, J. J.; SRIDHARAN, N. CAD-CAM integration using machining features. **Computer Integrated Manufacturing**, 15, n. 4, 2002. 296-318.

MINHAT, M. et al. A novel open CNC architecture based on STEP-NC data model and IEC 61499 function blocks. **Elsevier**, Nova Zelândia, v. 25, p. 560-569, Junho 2009.

NARAYAN, L.; RAO, M.; SARCAR, M. M. M. **Computer Aided Design And Manufacturing**. India: PHI Learning Pvt. Ltd. , 2008. 698 p.

NASSEHI, A.; NEWMAN, S. T.; ALEN, R. D. The application of multi-agent systems for STEP-NC computer aided process planning of prismatic components. **International Journal of Machine Tools & Manufacture** , Agosto 2005.

NIST SCL. **STEP Class Library (SCL), NIST, STEP Class Library Components**, 1999. Disponível em: <<http://www.nist.gov/el/msid/scl.cfm>>.

PAREDES, E. G. **Desenvolvimento de uma fresadora CNC aderente à norma STEP-NC baseado no controlador de máquina avançado (EMC2)**. UnB. Brasília, p. 202. 2013. (Dissertação de Mestrado em Sistemas Mecatrônicos).

PEREIRA, A. G. Desenvolvimento e avaliação de um editor para programação CN em centros de usinagem. **Dissertação de Mestrado em Engenharia Mecânica**, Curitiba, 2003. 122.

ROSSO JR, R. S. U. **STEP Compliant CAD/CAPP/CAM System For Rotational Asymmetric Parts. Thesis (Doctoral) - Wolfson School of Mechanical and Manufacturing Engineering**. Loughborough University. UK. 2005.

ROSSO JR, R. S. U.; NEWMAN, S. T. Estrutura de dados para sistemas CAD/CAM aderente à STEP. **Livro de Atas do VI Congresso Ibero Americano de Engenharia Mecânica**, Coimbra, p. 6, Outubro 2003.

SÄÄSKI, J.; SOLONEN, T.; PARO, J. **Integration of CAD, CAM and NC with Step-NC**. VTT Industrial Systems. Finlândia, p. 24. 2005.

SEBNF2PARS. **Google code, A tool for building STEP Part 21 file parsers**, 2009. Disponível em: <<http://code.google.com/p/sebnf2pars/>>.

SIVITTER, S. Usinagem baseada em modelos sólidos será padrão. **CADesign**, n. 10, p. 14-15, Março 2004.

SOUZA JÚNIOR, A. M. Estudos da Utilização de PCBN e Cerâmica Mista no Fresamento de Blocos de Motores de Ferro Fundido Cinzento. Belo Horizonte: [s.n.], 2001. p. 142. Dissertação (Mestrado em Engenharia Mecânica) – PUCMINAS.

STARK, J.; KHAI, N. V.; MAEDER, W. Standardisation of the Manufacturing Process : the IMS STEP-NC project. Available: http://www.step-nc.org/data/IPLnetWorkshop2002_paper_27082002.pdf. **ESPRIT Project**, Setembro 2002.

STEP Tools. **Step Tools**. Disponível em: <<http://www.steptools.com>>. Acesso em: 02 Outubro 2012.

SUH, H. et al. Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. **Computer-Aided Design**, p. 1069 - 1083, October 2003.

SUH, S. H. et al. Developing an integrated STEP-compliant CNC prototype. **Journal of Manufacturing Systems.**, p. 350, Maio 2002.

SUH, S. H. et al. **Theory and Design of CNC Systems (Springer Series in Advanced Manufacturing)**. Londres: Springer, 2008. 456 p.

TANAKA, ; KISHINAMI, . STEP-based quality diagnosis of shape data of product models for collaborative e-engineering. **Advanced Computer Support of Engineering and Service Processes of Virtual Enterprises - Advanced Computer Support Special Issue.**, 57, 2006. 245-260.

WOSNIK, M.; KRAMER, C.; KLEMM, P. Enabling feedback of process data by use of STEP-NC. **International Journal of Computer Integrated Manufacturing.**, p. 559-569, Dezembro 2009.

XU, X. W. et al. STEP-compliant NC research: the search for intelligent CAD/CAPP/CAM/CNC integration. **International Journal of Production Research.**, Nova Zelândia, Reino Unido, v. 43, Setembro 2005. ISSN 17.

XU, X.; HE, Q. **Striving for a total integration of CAD, CAPP, CAM and CNC**. [S.l.]: Robotics and Computer-Integrated Manufacturing., v. 20, 2004. 101-109 p.

YANG, W.; XU, X. Modelling machine tool data in support of STEP-NC based manufacturing. **International Journal of Computer Integrated Manufacturing.**, Nova Zelândia, p. 745-763, Dezembro 2009.

YUSOF, Y.; TAN, Z. Z.; KASSIM, . ISO14649 Code Generator for Intelligent Manufacture for STEP-NC Compliant Machining. **Proceedings of the Asia**

Pacific industrial engineering & management systems conference (APIEMS2009)., Malaysia, 14-16 Dezembro 2009.

YUSRI , Y. ISO 14649 (STEP-NC): New Standards for CNC Machining. **International Journal of Integrated Engineering (Issue on Mechanical, Materials and Manufacturing Engineering)**., Malásia, v. 2, n. 1, 2010.

ZHANG, X. et al. A STEP-compliant process planning system for CNC turning operations. **Elsevier**, Reino Unido, China, Julho 2010.

APÊNDICE A - CONFIGURAÇÃO DE PINOS DA PLACA CNC USB CONTROLLER MK1

Este apêndice fornece informações complementares em relação à configuração da placa CNC USB Controller.

A Figura 46 mostra os conectores do MK1 Motor responsáveis pelo controle e envio dos sinais de pulso e direção para o motor de passo. Os pinos 2,4,6,7,8 e 10 fornecem o *ground* (terra) ou comum de conexões.



Figura 46 - MK1 Motor Connector.

Fonte: Elaborado pelo autor.

O conector MK1 conforme a Figura 47, permite o controle de dispositivos externos que podem ser incorporados no controlador. Os pinos 2,4,6,8,10,12,14 e 16 fornecem o *ground* (terra) ou comum de conexões.

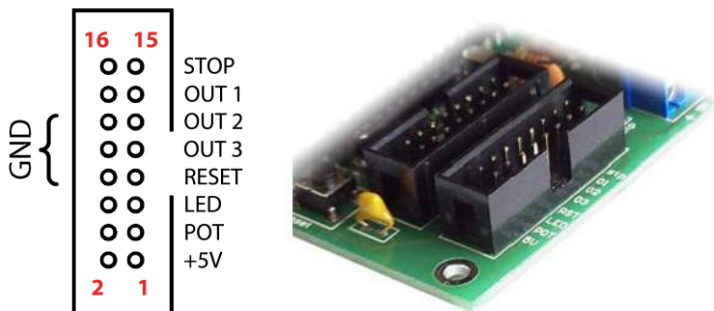


Figura 47 - MK1 Connector.

Fonte: Elaborado pelo autor.

O conector *jog* é controlado pelo funcionamento alternado dos *jogs* de 1 a 8, conforme a Figura 48. Um *plug-in* opcional (*jogging keyboard*) pode ser fornecido. Se os usuários construírem seu próprio teclado *jogging*, é recomendável que um capacitor de 100nF seja conectado entre os terminais do interruptor. Os pinos 2,4,6,8,10,12,14 e 16 são conexões contendo tensões de +5V.

- JOG 2: Jog eixo X na direção positiva.
- JOG 1: Jog eixo X na direção negativa.
- JOG 4: Jog eixo Y na direção positiva.
- JOG 3: Jog eixo Y na direção negativa.
- JOG 6: Jog eixo Z na direção positiva.
- JOG 5: Jog eixo Z na direção negativa.
- JOG 8: Jog eixo A na direção positiva.
- JOG 7: Jog eixo A na direção negativa.

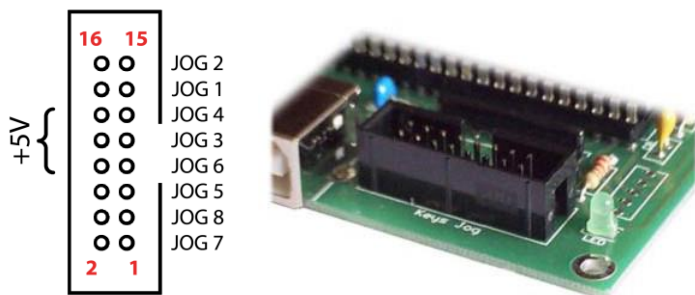


Figura 48 - MK1 JOG Connector.

Fonte: Elaborado pelo autor.

Os *limits* de 1 a 8 (Figura 49) conectam chaves fim de curso. Recomenda-se que um capacitor de 100nF seja conectado diretamente entre os terminais do interruptor. Os pinos 2,4,6,8,10,12,14 e 16 são *ground* (terra) ou conexões comuns.

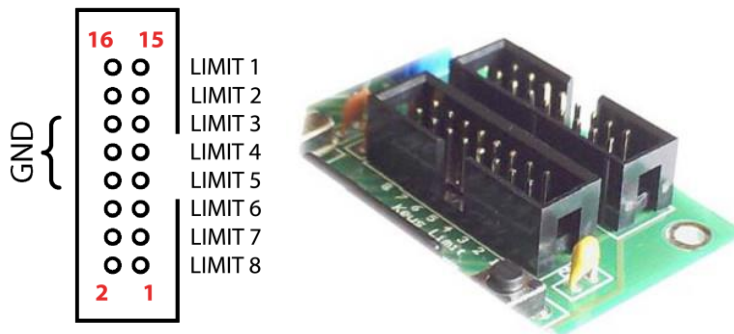


Figura 49 - MK1 LIMIT Connector.

Fonte: Elaborado pelo autor.

Abaixo segue a lista de limites para os eixos X,Y e Z:

- LIMITE 1: Limite negativo do eixo X.
- LIMITE 2: Limite positivo do eixo X.
- LIMITE 3: Limite negativo do eixo Y.
- LIMITE 4: Limite positivo do eixo Y.
- LIMITE 5: Limite negativo do eixo Z.
- LIMITE 6: Limite positivo do eixo Z.
- LIMITE 7: Limite negativo do eixo A.
- LIMITE 8: Limite positivo do eixo A.

APÊNDICE B - DOWNLOAD E UPLOAD DO ARQUIVO “.HEX” NO ARDUINO MEGA2560

O AVR *Downloader Uploader*, também conhecido como AVRDUDE (Figura 50), é um programa para *download* e *upload* das memórias *on-chip* de microcontroladores AVR da Atmel. Pode-se programar o *Flash* e EEPROM quando suportado pelo protocolo de programação serial, e ele pode combinar o programa em pedaços e causar o bloqueio do mesmo. O AVRDUDE também fornece um modo de instrução direta, permitindo que se emita qualquer instrução de programação para o chip AVR, e independentemente o AVRDUDE implementa essa característica específica em um *chip* especial.

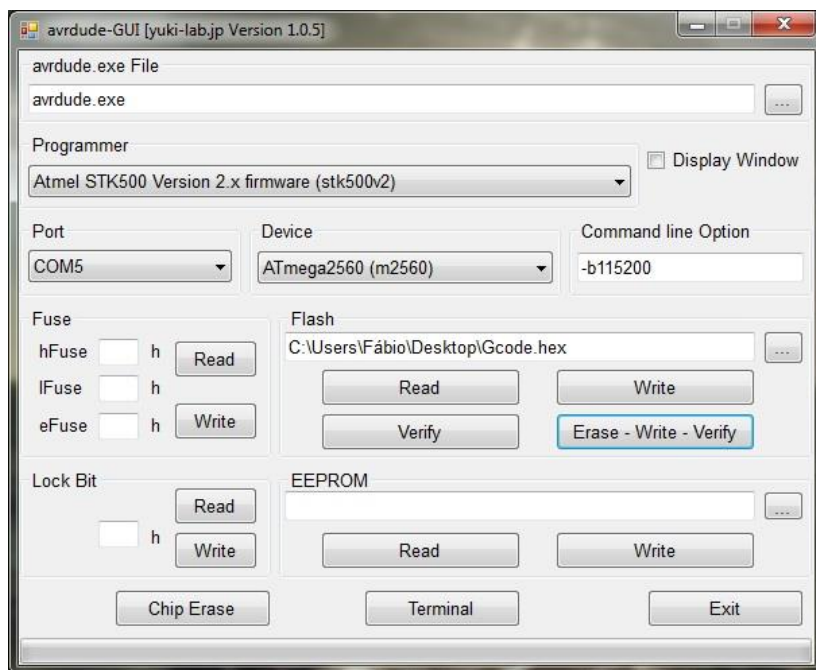


Figura 50 - Configuração de *upload* do arquivo “.hex” no Arduino Mega2560.

Fonte: Elaborado pelo autor.

O *software* AVRDUDE pode ser utilizado de forma eficaz mediante linhas de comando para ler ou escrever em todos os tipos de *chips* de memória (EEPROM, *flash*, *bits* de fusíveis, *bits*, *bytes*) ou mediante um modo interativo (*terminal*). Usando AVRDUDE a partir da linha de comando, ele funciona bem para a programação de toda a memória do *chip* a partir do conteúdo de um arquivo, enquanto que no modo interativo é útil para explorar o conteúdo da memória, modificando *bytes* individuais de EEPROM, permite programação de fusíveis, *lock bits*, etc.

Ele também suporta os seguintes tipos de programador básicos: STK500 da Atmel, dispositivos mkII AVRISP e AVRISP da Atmel, STK600 da Atmel, da Atmel JTAG ICE (ambos MKI e mkII, este último também no modo ISP), *appnote* avr910, *appnote* avr109 (incluindo o AVR *Butterfly*), adaptadores seriais *bit-bang*, e o PPI (interface de porta paralela). O PPI representa uma classe de programadores simples, onde as linhas de programação são diretamente conectadas à porta paralela do computador. Existem várias configurações de pinos para diversas variações de programadores PPI, e o *software* AVRDUDE pode ser configurado para trabalhar com eles, ou especificando o programador apropriado na linha de comando ou mediante a criação de uma nova entrada no seu arquivo de configuração. Tudo o que for necessário para uma nova entrada é considerado no *software* AVRDUDE, definindo-se quais os pinos que serão utilizados para cada função declarada na programação.

APÊNDICE C - ARQUIVO STEP-NC CONSTRUÍDO PARA O TESTE DE FRESAMENTO RANHURA.

```

ISO-10303-21;
HEADER;
/*GRIMA*/
FILE_DESCRIPTION(
/* description */ ('Descricao'),
/* implementation_level */ ('2;1');
FILE_NAME(
/* name */ ('',
/* time_stamp */ ('2013-11-26T17:22:36',
/* author */ ('fabioautoind'),
/* organization */ ('UFSC'),
/* preprocessor_version */ ('ST-GENERATOR 5.3',
/* originating_system */ ('',
/* authorization */ (''));
FILE_SCHEMA(('MACHINING_SCHEMA'));
ENDSEC;
DATA;
#1=PROJECT('Fresamento Ranhura',#70,(#4),#2,$,$);
#2=PERSON_AND_ADDRESS(#3,$);
#3=PERSON($,$,'fabioautoind',$,$);
#4=WORKPIECE('workpiece',#9,50.0,$,$,#11,());
#5=AXIS2_PLACEMENT_3D('workpiece placement',#6,#7,#8);
#6=CARTESIAN_POINT("(, (0.0,0.0,0.0));
#7=DIRECTION("(, (0.0,0.0,1.0));
#8=DIRECTION("(, (1.0,0.0,0.0));
#9=MATERIAL('SAE 1020','ACO SEM LIGA',(#10));
#10=NUMERIC_PARAMETER('Hardness',250.0,'HB');
#11=BLOCK('piece',#5,100.0,50.0,10.0);
#12=SETUP('setup',#18,#17,(#22));
#13=AXIS2_PLACEMENT_3D("#, #14, #15, #16);
#14=CARTESIAN_POINT("(, (0.0,0.0,35.0));
#15=DIRECTION("(, (0.0,0.0,1.0));
#16=DIRECTION("(, (1.0,0.0,0.0));
#17=PLANE('security plane',#13);
#18=AXIS2_PLACEMENT_3D('origin',#19,#20,#21);
#19=CARTESIAN_POINT("(, (0.0,0.0,0.0));
#20=DIRECTION("(, (0.0,0.0,1.0));
#21=DIRECTION("(, (1.0,0.0,0.0));
#22=WORKPIECE_SETUP(#4,#23,$,$);
#23=AXIS2_PLACEMENT_3D("#, #24, #25, #26);
#24=CARTESIAN_POINT("(, (0.0,0.0,0.0));

```

```

#25=DIRECTION(",(0.0,0.0,1.0));
#26=DIRECTION(",(1.0,0.0,0.0));
#27=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side
Rough Milling',5.0,#35,#28,#33,#34,$,#36,#37,#40,2.5,6.0,0.0,0.0);
#28=MILLING_CUTTING_TOOL('6mmGiacometCNC',#29,(#31),$,$,$);
#29=FACEMILL(#30,2,.,RIGHT.,$,40.0);
#30=TOOL_DIMENSION(6.0,0.0,0.0,20.0,0.0,0.0,0.0);
#31=CUTTING_COMPONENT(60.0,#32,$,$,$);
#32=MATERIAL('P',$,$);
#33=MILLING_TECHNOLOGY(1000.0,.,TCP.,900.0,7200.0,$,F,.,F,.,$);
#34=MILLING_MACHINE_FUNCTIONS(.,T.,$,.,T.,$,.,T.,$,,$,$);
#35=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#36=PLUNGE_TOOLAXIS(#38);
#37=PLUNGE_TOOLAXIS(#39);
#38=DIRECTION('approach strategy direction',(0.0,0.0,1.0));
#39=DIRECTION('retract strategy direction',(0.0,0.0,-1.0));
#40=CONTOUR_PARALLEL(1.2000000000000002,.,T.,.,CCW,.,CONVENTI
ONAL.);
#41=SLOT('ranhura',#4,(#27),#42,#50,#55,#59,(#66,#67));
#42=AXIS2_PLACEMENT_3D('ranhura placement',#43,#44,#45);
#43=CARTESIAN_POINT(",(0.0,25.0,10.0));
#44=DIRECTION(",(0.0,0.0,1.0));
#45=DIRECTION(",(1.0,0.0,0.0));
#46=AXIS2_PLACEMENT_3D('Slot depth',#47,#48,#49);
#47=CARTESIAN_POINT(",(0.0,0.0,-5.0));
#48=DIRECTION(",(0.0,0.0,1.0));
#49=DIRECTION(",(1.0,0.0,0.0));
#50=PLANE('ranhura depth location',#46);
#51=AXIS2_PLACEMENT_3D('course of travel placement',#52,#53,#54);
#52=CARTESIAN_POINT(",(0.0,0.0,0.0));
#53=DIRECTION(",(0.0,0.0,1.0));
#54=DIRECTION(",(1.0,0.0,0.0));
#55=LINEAR_PATH(#51,#56,#58);
#56=TOLERANCED_LENGTH_MEASURE(100.0,#57);
#57=PLUS_MINUS_VALUE(0.05,0.05,4);
#58=DIRECTION('Linear Path Direction - HORIZONTAL',(1.0,0.0,0.0));
#59=SQUARE_U_PROFILE(#62,#60,$,$,$,$);
#60=TOLERANCED_LENGTH_MEASURE(6.0,#61);
#61=PLUS_MINUS_VALUE(0.05,0.05,4);
#62=AXIS2_PLACEMENT_3D($,#63,#64,#65);
#63=CARTESIAN_POINT(",(0.0,0.0,-5.0));

```



```
#63=CARTESIAN_POINT("(0.0,0.0,-5.0));  
#64=DIRECTION("(0.0,0.0,1.0));  
#65=DIRECTION("(0.0,-1.0,0.0));  
#66=OPEN_SLOT_END_TYPE();  
#67=OPEN_SLOT_END_TYPE();  
#68=MACHINING_WORKINGSTEP('Slot_RGH',#17,#41,#27,$);  
#69=WORKPLAN('workplan setup XY',(#68),$,#12,$);  
#70=WORKPLAN('Main workplan',(#69),$,$,$);  
ENDSEC;  
END-ISO-10303-21;
```


**APÊNDICE D - CÓDIGO G CONSTRUÍDO PARA O TESTE DE
FRESAMENTO RANHURA.**

G54
G90
S7200.0 F1000.0 M3
M8
G1 X0.0 Y22.0 Z-0.0
G1 X0.0 Y22.0 Z-2.5
G1 X100.0 Y22.0 Z-2.5
G1 X100.0 Y28.0 Z-2.5
G1 X0.0 Y28.0 Z-2.5
G1 X0.0 Y28.0 Z-5.0
G1 X100.0 Y28.0 Z-5.0
G1 X100.0 Y22.0 Z-5.0
G1 X0.0 Y22.0 Z-5.0
G1 X0.0 Y22.0 Z-0.0
G0 Z5.0
M30

APÊNDICE E - ARQUIVO STEP-NC CONSTRUÍDO PARA TESTE DE FURAÇÃO E FRESAMENTO CIRCULAR.

```

ISO-10303-21;
HEADER;
/*GRIMA*/
FILE_DESCRIPTION(
/* description */ ('Descricao'),
/* implementation_level */ ('2;1');
FILE_NAME(
/* name */ (''),
/* time_stamp */ ('2013-12-05T18:47:13',
/* author */ ('fabioautoind'),
/* organization */ ('UFSC'),
/* preprocessor_version */ ('ST-GENERATOR 5.3',
/* originating_system */ (''),
/* authorization */ (''));
FILE_SCHEMA(('MACHINING_SCHEMA'));
ENDSEC;
DATA;
#1=PROJECT('Furação e Fresamento Circular STEP-NC',#218,(#4),#2,$,$);
#2=PERSON_AND_ADDRESS(#3,$);
#3=PERSON($,$,'fabioautoind',$,$);
#4=WORKPIECE('workpiece',#9,50.0,$,$,#11,());
#5=AXIS2_PLACEMENT_3D('workpiece placement',#6,#7,#8);
#6=CARTESIAN_POINT("(, (0.0,0.0,0.0));
#7=DIRECTION("(, (0.0,0.0,1.0));
#8=DIRECTION("(, (1.0,0.0,0.0));
#9=MATERIAL('SAE 1020','ACO SEM LIGA',#10));
#10=NUMERIC_PARAMETER('Hardness',250.0,'HB');
#11=BLOCK('piece',#5,150.0,100.0,30.0);
#12=SETUP('setup',#18,#17,(#22));
#13=AXIS2_PLACEMENT_3D("#14,#15,#16);
#14=CARTESIAN_POINT("(, (0.0,0.0,55.0));
#15=DIRECTION("(, (0.0,0.0,1.0));
#16=DIRECTION("(, (1.0,0.0,0.0));
#17=PLANE('security plane',#13);
#18=AXIS2_PLACEMENT_3D('origin',#19,#20,#21);
#19=CARTESIAN_POINT("(, (0.0,0.0,0.0));
#20=DIRECTION("(, (0.0,0.0,1.0));
#21=DIRECTION("(, (1.0,0.0,0.0));
#22=WORKPIECE_SETUP(#4,#23,$,$);
#23=AXIS2_PLACEMENT_3D("#24,#25,#26);
#24=CARTESIAN_POINT("(, (0.0,0.0,0.0));

```

```

#25=DIRECTION(",(0.0,0.0,1.0));
#26=DIRECTION(",(1.0,0.0,0.0));
#27=CENTER_DRILLING($,$,'Center
Drilling',5.0,#28,#29,#34,#35,$,5.0,$,$,#36);
#28=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#29=MILLING_CUTTING_TOOL('center drill',#30,(#32),$,$,$);
#30=CENTER_DRILL(#31,2.,.RIGHT.,$,25.0);
#31=TOOL_DIMENSION(3.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0);
#32=CUTTING_COMPONENT(50.0,#33,$,$,$);
#33=MATERIAL('P',$,$);
#34=MILLING_TECHNOLOGY(200.0.,TCP.,110.0,7200.0,$.F.,F.,F.,$);
#35=MILLING_MACHINE_FUNCTIONS(.T.,$,$.T.,$,$.T.,$,$,$);
#36=DRILLING_TYPE_STRATEGY($,$,$,$,$);
#37=ROUND_HOLE('Furo com base conica',#4,(#27,#51),#38,#46,#47,$,#49);
#38=AXIS2_PLACEMENT_3D('Furo          com          base          conica
placement',#39,#40,#41);
#39=CARTESIAN_POINT(",(25.0,20.0,30.0));
#40=DIRECTION(",(0.0,0.0,1.0));
#41=DIRECTION(",(1.0,0.0,0.0));
#42=AXIS2_PLACEMENT_3D('hole depth',#43,#44,#45);
#43=CARTESIAN_POINT(",(0.0,0.0,-25.0));
#44=DIRECTION(",(0.0,0.0,1.0));
#45=DIRECTION(",(1.0,0.0,0.0));
#46=PLANE('hole depth plane',#42);
#47=TOLERANCED_LENGTH_MEASURE(6.0,#48);
#48=PLUS_MINUS_VALUE(0.05,0.05,4);
#49=CONICAL_HOLE_BOTTOM(0.3490658503988659,$);
#50=MACHINING_WORKINGSTEP('ConicalBottomHole1.0_RGH',#17,#37,#
27,$);
#51=DRILLING($,$,'Drilling',5.0,#59,#52,#57,#58,$,25.0,$,$,#60);
#52=MILLING_CUTTING_TOOL('twist drill',#53,(#55),$,$,$);
#53=TWIST_DRILL(#54,2.,.LEFT.,$,60.0);
#54=TOOL_DIMENSION(6.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#55=CUTTING_COMPONENT(70.0,#56,$,$,$);
#56=MATERIAL('P',$,$);
#57=MILLING_TECHNOLOGY(300.0.,TCP.,110.0,7200.0,$.F.,F.,F.,$);
#58=MILLING_MACHINE_FUNCTIONS(.T.,$,$.T.,$,$.T.,$,$,$);
#59=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#60=DRILLING_TYPE_STRATEGY($,$,$,$,$);
#61=MACHINING_WORKINGSTEP('ConicalBottomHole1.1_RGH',#17,#37,#
51,$);

```

```

#62=CENTER_DRILLING($,$,'Center
Drilling',5.0,#63,#64,#69,#70,$,5.0,$,$,#71);
#63=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#64=MILLING_CUTTING_TOOL('center drill',#65,(#67),$,$,$);
#65=CENTER_DRILL(#66,2,.,RIGHT,.,$,25.0);
#66=TOOL_DIMENSION(3.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0);
#67=CUTTING_COMPONENT(50.0,#68,$,$,$);
#68=MATERIAL('P',$,$);
#69=MILLING_TECHNOLOGY(200.0,.,TCP,.,110.0,7200.0,$,.,F,.,F,.,F,.,$);
#70=MILLING_MACHINE_FUNCTIONS(.,T,.,$,.,T,.,$,.,T,.,$,.,$);
#71=DRILLING_TYPE_STRATEGY($,$,$,$,$);
#72=ROUND_HOLE('Furo com base conica',#4,(#62,#86),#73,#81,#82,$,#84);
#73=AXIS2_PLACEMENT_3D('Furo com base conica
placement',#74,#75,#76);
#74=CARTESIAN_POINT(",(25.0,80.0,30.0));
#75=DIRECTION(",(0.0,0.0,1.0));
#76=DIRECTION(",(1.0,0.0,0.0));
#77=AXIS2_PLACEMENT_3D('hole depth',#78,#79,#80);
#78=CARTESIAN_POINT(",(0.0,0.0,-25.0));
#79=DIRECTION(",(0.0,0.0,1.0));
#80=DIRECTION(",(1.0,0.0,0.0));
#81=PLANE('hole depth plane',#77);
#82=TOLERANCED_LENGTH_MEASURE(6.0,#83);
#83=PLUS_MINUS_VALUE(0.05,0.05,4);
#84=CONICAL_HOLE_BOTTOM(0.3490658503988659,$);
#85=MACHINING_WORKINGSTEP('ConicalBottomHole2.0_RGH',#17,#72,#
62,$);
#86=DRILLING($,$,'Drilling',5.0,#94,#87,#92,#93,$,25.0,$,$,#95);
#87=MILLING_CUTTING_TOOL('twist drill',#88,(#90),$,$,$);
#88=TWIST_DRILL(#89,2,.,LEFT,.,$,60.0);
#89=TOOL_DIMENSION(6.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#90=CUTTING_COMPONENT(70.0,#91,$,$,$);
#91=MATERIAL('P',$,$);
#92=MILLING_TECHNOLOGY(400.0,.,TCP,.,110.0,7200.0,$,.,F,.,F,.,F,.,$);
#93=MILLING_MACHINE_FUNCTIONS(.,T,.,$,.,T,.,$,.,T,.,$,.,$);
#94=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#95=DRILLING_TYPE_STRATEGY($,$,$,$,$);
#96=MACHINING_WORKINGSTEP('ConicalBottomHole2.1_RGH',#17,#72,#
86,$);
#97=CENTER_DRILLING($,$,'Center
Drilling',5.0,#98,#99,#104,#105,$,5.0,$,$,#106);

```

```

#98=CARTESIAN_POINT('start point',(0,0,0,0,0));
#99=MILLING_CUTTING_TOOL('center drill',#100,(#102),$,,$);
#100=CENTER_DRILL(#101,2,RIGHT,,$,25.0);
#101=TOOL_DIMENSION(3.0,0.5235987755982988,0,0,10.0,0,0,0,0,0);
#102=CUTTING_COMPONENT(50.0,#103,$,$,$);
#103=MATERIAL('P',,$,$);
#104=MILLING_TECHNOLOGY(200.0,TCP,110.0,7200.0,$,F..F..F.,$);
#105=MILLING_MACHINE_FUNCTIONS(T.,$,,$,T.,$,,$,T.,$,,$,$);
#106=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#107=ROUND_HOLE('Furo com base
conica',#4,(#97,#121),#108,#116,#117,$,#119);
#108=AXIS2_PLACEMENT_3D('Furo com base conica
placement',#109,#110,#111);
#109=CARTESIAN_POINT(",(125.0,20.0,30.0));
#110=DIRECTION(",(0.0,0.0,1.0));
#111=DIRECTION(",(1.0,0.0,0.0));
#112=AXIS2_PLACEMENT_3D('hole depth',#113,#114,#115);
#113=CARTESIAN_POINT(",(0.0,0.0,-25.0));
#114=DIRECTION(",(0.0,0.0,1.0));
#115=DIRECTION(",(1.0,0.0,0.0));
#116=PLANE('hole depth plane',#112);
#117=TOLERANCED_LENGTH_MEASURE(6.0,#118);
#118=PLUS_MINUS_VALUE(0.05,0.05,4);
#119=CONICAL_HOLE_BOTTOM(0.3490658503988659,$);
#120=MACHINING_WORKINGSTEP('ConicalBottomHole3.0_RGH',#17,#10
7,#97,$);
#121=DRILLING($,$,'Drilling',5.0,#129,#122,#127,#128,$,25.0,$,$,$,#130);
#122=MILLING_CUTTING_TOOL('twist drill',#123,(#125),$,,$,$);
#123=TWIST_DRILL(#124,2,LEFT,,$,60.0);
#124=TOOL_DIMENSION(6.0,0.3490658503988659,0,0,40.0,0,0,0,0,0);
#125=CUTTING_COMPONENT(70.0,#126,$,$,$);
#126=MATERIAL('P',,$,$);
#127=MILLING_TECHNOLOGY(500.0,TCP,110.0,7200.0,$,F..F..F.,$);
#128=MILLING_MACHINE_FUNCTIONS(T.,$,,$,T.,$,,$,T.,$,,$,$);
#129=CARTESIAN_POINT('start point',(0,0,0,0,0));
#130=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#131=MACHINING_WORKINGSTEP('ConicalBottomHole3.1_RGH',#17,#10
7,#121,$);
#132=CENTER_DRILLING($,$,'Center
Drilling',5.0,#133,#134,#139,#140,$,5.0,$,$,$,#141);
#133=CARTESIAN_POINT('start point',(0,0,0,0,0));

```



```

#134=MILLING_CUTTING_TOOL('center drill',#135,(#137),$,,$);
#135=CENTER_DRILL(#136,2,RIGHT,,$,25.0);
#136=TOOL_DIMENSION(3.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0);
#137=CUTTING_COMPONENT(50.0,#138,$,$);
#138=MATERIAL('P',,$);
#139=MILLING_TECHNOLOGY(200.0,TCP,110.0,7200.0,$,F,.,F,.,F,.);
#140=MILLING_MACHINE_FUNCTIONS(.T,,$,.,T,,$,.,T,,$,,$);
#141=DRILLING_TYPE_STRATEGY($,$,$,$,$);
#142=ROUND_HOLE('Furo com base
conica',#4,(#132,#156),#143,#151,#152,$,#154);
#143=AXIS2_PLACEMENT_3D('Furo com base conica
placement',#144,#145,#146);
#144=CARTESIAN_POINT(",(125.0,80.0,30.0));
#145=DIRECTION(",(0.0,0.0,1.0));
#146=DIRECTION(",(1.0,0.0,0.0));
#147=AXIS2_PLACEMENT_3D('hole depth',#148,#149,#150);
#148=CARTESIAN_POINT(",(0.0,0.0,-25.0));
#149=DIRECTION(",(0.0,0.0,1.0));
#150=DIRECTION(",(1.0,0.0,0.0));
#151=PLANE('hole depth plane',#147);
#152=TOLERANCED_LENGTH_MEASURE(6.0,#153);
#153=PLUS_MINUS_VALUE(0.05,0.05,4);
#154=CONICAL_HOLE_BOTTOM(0.3490658503988659,$);
#155=MACHINING_WORKINGSTEP('ConicalBottomHole4.0_RGH',#17,#14
2,#132,$);
#156=DRILLING($,$,'Drilling',5.0,#164,#157,#162,#163,$,25.0,$,$,$,#165);
#157=MILLING_CUTTING_TOOL('twist drill',#158,(#160),$,,$);
#158=TWIST_DRILL(#159,2,.,LEFT,.,$,60.0);
#159=TOOL_DIMENSION(6.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#160=CUTTING_COMPONENT(70.0,#161,$,$);
#161=MATERIAL('P',,$);
#162=MILLING_TECHNOLOGY(600.0,TCP,110.0,7200.0,$,F,.,F,.,F,.);
#163=MILLING_MACHINE_FUNCTIONS(.T,,$,.,T,,$,.,T,,$,,$);
#164=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#165=DRILLING_TYPE_STRATEGY($,$,$,$,$);
#166=MACHINING_WORKINGSTEP('ConicalBottomHole4.1_RGH',#17,#14
2,#156,$);
#167=CENTER_DRILLING($,$,'Center
Drilling',5.0,#168,#169,#174,#175,$,5.0,$,$,$,#176);
#168=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#169=MILLING_CUTTING_TOOL('center drill',#170,(#172),$,,$);

```

```

#170=CENTER_DRILL(#171,2,.RIGHT.,$,25.0);
#171=TOOL_DIMENSION(3.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0);
#172=CUTTING_COMPONENT(50.0,#173,$,$,$);
#173=MATERIAL('P',$,$);
#174=MILLING_TECHNOLOGY(200.0,.TCP.,110.0,7200.0,$.F..F..F.,$);
#175=MILLING_MACHINE_FUNCTIONS(.T.,$,$.T.,$,$.T.,$,$,$);
#176=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#177=ROUND_HOLE('Furo com base
plana',#4,(#167,#191,#202),#178,#188,#182,$,#189);
#178=AXIS2_PLACEMENT_3D('Furo com base plana
placement',#179,#180,#181);
#179=CARTESIAN_POINT(",(75.0,50.0,30.0));
#180=DIRECTION(",(0.0,0.0,1.0));
#181=DIRECTION(",(1.0,0.0,0.0));
#182=TOLERANCED_LENGTH_MEASURE(72.0,#183);
#183=PLUS_MINUS_VALUE(0.05,0.05,4);
#184=AXIS2_PLACEMENT_3D('hole depth',#185,#186,#187);
#185=CARTESIAN_POINT(",(0.0,0.0,-10.0));
#186=DIRECTION(",(0.0,0.0,1.0));
#187=DIRECTION(",(1.0,0.0,0.0));
#188=PLANE('hole depth plane',#184);
#189=FLAT_HOLE_BOTTOM();
#190=MACHINING_WORKINGSTEP('FlatBottomHole5.0_RGH',#17,#177,#1
67,$);
#191=DRILLING($,$,'Drilling',5.0,#199,#192,#197,#198,$,15.0,$,$,#200);
#192=MILLING_CUTTING_TOOL('twist drill',#193,(#195),$,$,$);
#193=TWIST_DRILL(#194,2,.LEFT.,$,60.0);
#194=TOOL_DIMENSION(6.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#195=CUTTING_COMPONENT(70.0,#196,$,$,$);
#196=MATERIAL('P',$,$);
#197=MILLING_TECHNOLOGY(700.0,.TCP.,110.0,7200.0,$.F..F..F.,$);
#198=MILLING_MACHINE_FUNCTIONS(.T.,$,$.T.,$,$.T.,$,$,$);
#199=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#200=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#201=MACHINING_WORKINGSTEP('FlatBottomHole5.1_RGH',#17,#177,#1
91,$);
#202=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side
Rough Milling',5.0,#210,#203,#208,#209,$,#211,#212,#215,10.0,36.0,0.0,0.0);
#203=MILLING_CUTTING_TOOL('6mm CNC UFSC',#204,(#206),$,$,$);
#204=FACEMILL(#205,2,.RIGHT.,$,60.0);
#205=TOOL_DIMENSION(6.0,0.0,0.0,40.0,0.0,0.0,0.0);

```

```

#206=CUTTING_COMPONENT(70.0,#207,$,$);
#207=MATERIAL('P',$,$);
#208=MILLING_TECHNOLOGY(1000.0,.TCP.,145.0,7.2007692E7,$,F.,F.,F
.,$);
#209=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,$.T.,$,$,$);
#210=CARTESIAN_POINT('start point',(0.0,3.0,0.0));
#211=PLUNGE_TOOLAXIS(#213);
#212=PLUNGE_TOOLAXIS(#214);
#213=DIRECTION('approach strategy direction',(0.0,0.0,1.0));
#214=DIRECTION('retract strategy direction',(0.0,0.0,-1.0));
#215=CONTOUR_PARALLEL(1.2000000000000002,.T.,CCW..CONVENTI
ONAL.);
#216=MACHINING_WORKINGSTEP('FlatBottomHole5.2_RGH',#17,#177,#2
02,$);
#217=WORKPLAN('workplan                                setup
XY',(#50,#61,#85,#96,#120,#131,#155,#166,#190,#201,#216),$,#12,$);
#218=WORKPLAN('Main workplan',(#217),$,$);
ENDSEC;
END-ISO-10303-21;

```


**APÊNDICE F - CÓDIGO G CONSTRUÍDO PARA O TESTE DE
FRESAMENTO RANHURA.**

```
G54
S7200.0 F90.0 M4
T = "center drill"
M6
G0 X115.0 Y80.0
M8
CYCLE81(5.0, -0.0, 5.0, -3.0, )
M9
S7350.0 F115.0 M4
T = "twist drill"
M6
G0 X115.0 Y80.0
CYCLE81(5.0, -0.0, 5.0, -10.0, )
S7500.0 F500.0 M3
T = "Fresa6mm-UFSC"
M6
G0 X115.0 Y80.0
M8
POCKET2(5.0, -0.0, , -10.0, , 12.0, 115.0, 80.0, 500.0, 1000.0, 5.0, 2.0, , , , )
M9
M30
```