



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

ESCOLA DE ENGENHARIA

TRABALHO DE CONCLUSÃO EM ENGENHARIA DE  
CONTROLE E AUTOMAÇÃO

# Projeto de Fresadora CNC com Plataforma Livre Arduino

*Autor: Alan Fachim*

*Orientador: Rafael A.C. Laranja*

Porto Alegre, Dezembro de 2013

## Sumário

RESUMO	iii
Lista de Figuras	iv
Lista de Tabelas	v
1 Introdução	1
2 Objetivos	2
3 Revisão Bibliográfica	3
<b>3.1</b> Medição de Incertezas	3
3.1.1    Exatidão e Repetitividade	3
3.1.2    Erros	4
3.1.3    Incerteza de Medição	4
<b>3.2</b> Hierarquia de erros em máquinas	6
<b>3.3</b> Precisão geométrica da máquina	6
3.3.1    Ensaio geométricos	6
3.3.2    Ensaio de Repetitividade de Posicionamento	7
3.3.3    Ensaio de Perpendicularidade	7
3.3.4    Ensaio de Retilidade	8
4 Metodologia	9
<b>4.1</b> Projeto Mecânico da Máquina	9
4.1.1    Seleção dos Componentes Mecânicos	11
4.1.2    Simulação e verificação do modelo	13
<b>4.2</b> Projeto Elétrico	14
4.2.1    Seleção dos atuadores	14
4.2.2    Seleção dos Drivers de potência	15
<b>4.3</b> Projeto do software, hardware e protocolo de comunicação	17
4.3.1    Projeto do software	17
4.3.2    Seleção do Hardware	23
4.3.3    Seleção do firmware do controlador e protocolo	23
5 Análise dos resultados	25
<b>5.1</b> Análise de incertezas	25
5.1.1    Análise de Repetitividade	25
5.1.2    Análise de Retilidade	27
5.1.3    Análise de Perpendicularidade e Paralelismo	29
6 Conclusões	31
<b>6.1</b> Trabalhos Futuros	32
7 Referências Bibliográficas	33
Apêndice A – Desenho da Estrutura Principal	34
Apêndice B – Custos	35
Anexo A – Drivers dos motores	36

## RESUMO

Este trabalho consiste na criação de um protótipo de fresadora controlada por Comando Numérico Computadorizado (CNC), buscando minimizar a tolerância de erros assim como atender objetivos como baixo custo de manufatura e construir um programa de livre acesso que possa facilitar o trabalho de usinagem. A fim de atender a estes objetivos, o trabalho foi dividido em 3 partes: projeto mecânico, projeto elétrico e projeto de *software*. A respeito do projeto mecânico, foi realizada a construção da estrutura mecânica da máquina através de processos de dobra e corte de chapas de aço, bem como processos de soldagem. Além disso, foi realizada a seleção de elementos de máquinas visando à otimização de custos e qualidade. Para o projeto elétrico, foram escolhidos motores de passo controlados por drivers PWM com micropasso. O firmware utilizado no Arduino é o Grbl, o interpretador de comandos de código aberto mais utilizado em projetos DIY (*do it yourself*), o qual se comunica com o Gstudio, um software que foi desenvolvido para fazer a interface com o usuário. O Gstudio permite a criação de desenhos, alteração do código G, e visualização em tempo real do funcionamento da máquina. Os resultados obtidos se mostraram satisfatórios, pois o protótipo funcionou de acordo com as características desejadas de minimização das tolerâncias e facilidade de operação.

## PALAVRAS CHAVES

Fresadora, CNC, Usinagem, Arduino, Grbl, Gstudio.

## ABSTRACT

This work presents the project of a milling machine prototype with Computer Numerical Control (CNC), with the goal to ally the low cost open platform Arduino with good precision, as well as user interface software to facilitate work machining. To meet these goals the work was divided on mechanical, electrical design and software project. The mechanical project aimed to use processes of bending and cutting of steel plates, and welding process for the main structure, and a careful selection of machine elements for optimizing cost and quality. The electrical project was based on stepper motors and use of PWM micro stepping drivers. As a user interface software, was developed a PC program with features like drawing, code alteration, and real-time machine monitoring, which communicates with the Arduino firmware. The firmware used is the Grbl, the most used G code interpreted in the DIY (do it yourself). The results were satisfactory as the prototype operated and matched the tolerance requirements and the software obtained a software interface to firmware in already well distributed control Grbl.

## KEYWORDS

Fresadora, CNC, Usinagem, Arduino, Grbl, Gstudio.

## Lista de Figuras

<b>Figura 1</b> - Exatidão e repetitividade. Fonte: CNI/COMPI, 2002. ....	3
<b>Figura 2</b> - Representação de uma medição.....	4
<b>Figura 3</b> - Distribuição normal. ....	5
<b>Figura 4</b> - Hierarquia de erros em máquinas. Fonte: autor. ....	6
<b>Figura 5</b> – Erros geométricos em guias de máquinas (STOETERAU) .....	6
<b>Figura 6</b> - Vista explodida da fresadora. Fonte: Autor.....	11
<b>Figura 7</b> - Rolamento linear <i>Pillow Block</i> – SMA16-UU – fonte: Kalatec, 2013a.....	12
<b>Figura 8</b> - Mancal para rolamentos de fixação do fuso. Fonte: Autor.....	13
<b>Figura 9</b> - Parafusos utilizados na fixação das partes mecânicas da fresadora CNC. Fonte Promaq, 2013. ....	13
<b>Figura 10</b> - Motor de passo e curva de torque. Figura redesenhada com base na figura do fabricante Sanyo. ....	15
<b>Figura 11</b> – Driver de potência TB6560AHQ. Construção do painel elétrico.....	16
<b>Figura 12</b> – Esquemático do sistema de hardware de controle CNC. ....	16
<b>Figura 13</b> – Diagrama de conexões entre o controlador e os drivers. ....	17
<b>Figura 14</b> – Hierarquia principal do software de interface (Gstudio). ....	18
<b>Figura 15</b> – Hierarquia de tópicos referente à comunicação entre Gstudio e Grbl. ....	19
<b>Figura 16</b> – Hierarquia de processos de interpretação de código G. Fonte: Autor. ....	20
<b>Figura 17</b> – Fluxograma da interface do Gstudio. Fonte: Autor. ....	20
<b>Figura 18</b> – Representação da matriz de calibração de duas dimensões.....	22
<b>Figura 19</b> – Tela principal do Gstudio. ....	22
<b>Figura 20</b> – Plataforma de controle Arduino Uno R3. Fonte: Electron-Shop. ....	23
<b>Figura 21</b> – Imagem do protótipo construído. ....	25
<b>Figura 22</b> – Análise de repetitividade dividida em 9 regiões. ....	26
<b>Figura 23</b> – Teste de repetitividade nas regiões 1, 2 e 3. Apresentam desvios.....	26
<b>Figura 24</b> - Teste de repetitividade nas regiões 4, 5 e 6. Apresentam desvios padrões de 5,1 $\mu\text{m}$ , 4,4 $\mu\text{m}$ e 5,8 $\mu\text{m}$ respectivamente.....	26
<b>Figura 25</b> – Teste de repetitividade nas regiões 7, 8 e 9. Apresentam desvios padrões de 6,2 $\mu\text{m}$ , 3,1 $\mu\text{m}$ e 5,2 $\mu\text{m}$ respectivamente.....	27
<b>Figura 26</b> – Máquina de medição tridimensional.....	27
<b>Figura 27</b> – Teste de retilidade do eixo x apresentou desvio padrão de 0,0342 mm. ....	28
<b>Figura 28</b> – Teste de retilidade do eixo y apresentou desvio padrão de 0,0268mm. ....	28
<b>Figura 29</b> – Imagem digitalizada da placa de furos utilizada para os testes.....	29
<b>Figura 30</b> – Retas obtidas por regressão linear. (a) Verticais (b) Horizontais.....	29

---

## Lista de Tabelas

<b>Tabela 1</b> – Comandos suportados pelo interpretador Grbl .....	24
<b>Tabela 2</b> - Análise de inclinação das retas no sentido x. ....	30
<b>Tabela 3</b> - Análise de inclinação das retas no sentido y em relação à média. ....	30
<b>Tabela B1</b> - Custos dos materiais.....	35



## 1 Introdução

Segundo Goodrich (2012), atualmente vivencia-se uma nova etapa da Revolução Industrial, também conhecida como “A Era das Máquinas Livres”. Neste âmbito, construir uma máquina CNC de baixo custo não é suficiente para incentivar a adesão de novos “inventores”, mas faz-se necessário construir uma plataforma que traga resultados rápidos, e obtidos de maneira mais simples que o mercado oferece atualmente. O autor relata, ainda, que o uso de plataformas de *hardware* livre, como Arduino, Raspberry e outras, são de fácil uso e simplificam a tarefa de automatização de processos. Apostar nesta forma de pensar é o que vem revolucionando a indústria de *hardware* livre. Pearce (2012) refere-se à plataforma Arduino como: “A beleza desta ferramenta é que é muito fácil de usar, e torna muito simples a tarefa de automatizar processos.”

O Arduino, que é vendido por menos de R\$ 100,00 no varejo (Pearce, 2012), pode controlar qualquer instrumento científico, até mesmo um contador Geiger, um osciloscópio, ou mesmo um sequenciador de DNA. Uma das aplicações que mais se destaca é o uso dessa plataforma para controlar impressoras 3D, as quais também são de *hardware* aberto (Reprap, 2013). Entretanto, apesar de uma máquina de prototipagem ter várias aplicações importantes, elas apresentam certas limitações no sentido de construir apenas protótipos sintetizados à base de polímeros, que possuem custos elevados e não muito acessíveis. Desta forma, torna-se viável construir uma máquina de usinagem controlada por tais plataformas, que consiga criar peças e protótipos de forma similar às impressoras 3D, criando inúmeras formas tridimensionais, utilizando materiais de fácil acesso.

Com o intuito de atender a esta demanda, o presente trabalho pode ser enquadrado como uma possível solução, já que se trata do projeto e construção de um protótipo de uma fresadora CNC. A seguir são apresentadas as etapas de construção do protótipo, que estão divididas em três pontos principais (projeto mecânico, projeto elétrico e projeto de *software*), onde serão abordadas as características de projeto e metodologia utilizadas. Por fim, será discutida a análise dos resultados obtidos pelo protótipo.

## 2 Objetivos

O objetivo geral do trabalho é desenvolver e construir um protótipo de uma fresadora CNC, bem como seu software de interface e comunicação.

Para alcançar o objetivo geral do trabalho, faz-se necessário atender aos seguintes objetivos específicos:

- Projeto mecânico da máquina;
- Projeto elétrico dos atuadores e dos *drivers* de potência;
- Projeto do *software*, *hardware* e protocolos de comunicação.



### 3 Revisão Bibliográfica

A fim de facilitar a leitura, alguns tópicos da revisão bibliográfica serão apresentados no capítulo de metodologia, sendo aqui apresentado apenas o tópico de análise de incertezas.

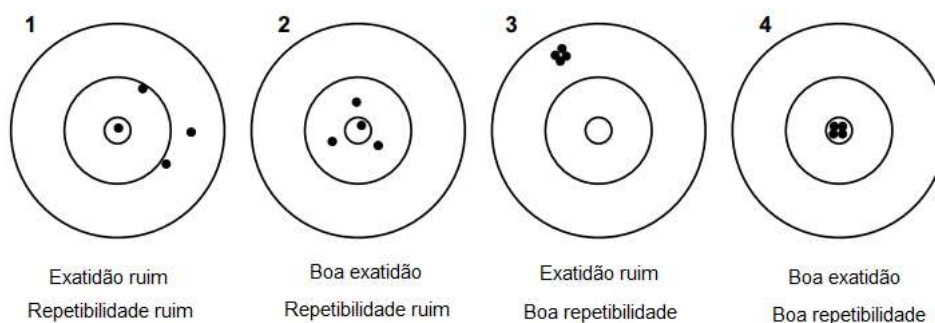
#### 3.1 Medição de Incertezas

Medir é um ato de comparar, e esta comparação envolve erros dos instrumentos, do operador, do processo de medida e outros. Logo, é intuitivo pensar que há a necessidade de se representar uma medida não somente através de um único valor, mas também por uma quantificação da qualidade dessa medição. Assim, deve-se adotar um valor que melhor represente a grandeza medida e uma margem de erro dentro da qual deve estar compreendido o valor real.

##### 3.1.1 Exatidão e Repetitividade

Em metrologia, os termos “exatidão” e “precisão” são considerados como características do processo de medição. A exatidão está associada à proximidade do valor verdadeiro, e a repetitividade está relacionada à dispersão dos valores resultantes de uma série de medidas.

Repetitividade é a capacidade de manter as medidas de uma peça para outra dentro de uma tolerância determinada pelo projeto (manter a maior similaridade possível entre as peças). A repetitividade de um sistema convencional vai se tornando mais difícil à medida que a complexidade vai aumentando. Um sistema onde praticamente tudo depende do operador não pode ser considerado repetitivo (SENAI\_RJ – Evolução do Comando Numérico, 1985).



**Figura 1** - Exatidão e repetitividade. Fonte: CNI/COMPI, 2002.

### 3.1.2 Erros

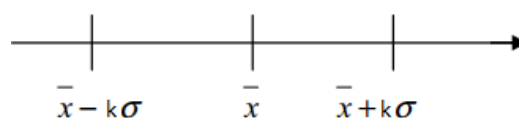
Segundo o Vocabulário Internacional de Metrologia de 2009 (VIM), os erros de medição são classificados em:

- Erro Grosseiro: É aquele cujo valor encontrado em conjuntos de medições difere dos outros. Os erros grosseiros, normalmente, correspondem a um valor que deve ser desprezado quando identificado e não deve ser tratado estatisticamente.
- Erro Sistemático: É a diferença entre a média de um determinado número de medições e o valor verdadeiro convencional. Este erro pode ser eliminado na calibração, pois normalmente ocorre em função de uma causa constante. Os erros sistemáticos fazem a média de um conjunto de medições se afastarem de um valor verdadeiro aceitável e afetam a exatidão dos resultados.
- Erro Aleatório: É a diferença entre o resultado de uma medição e a média de um determinado número de medições. Os erros aleatórios acontecem em função de causas irregulares e imprevisíveis e dificilmente podem ser eliminados. Os erros aleatórios ocasionam medições espalhadas de forma relativamente simétrica em torno do valor médio.

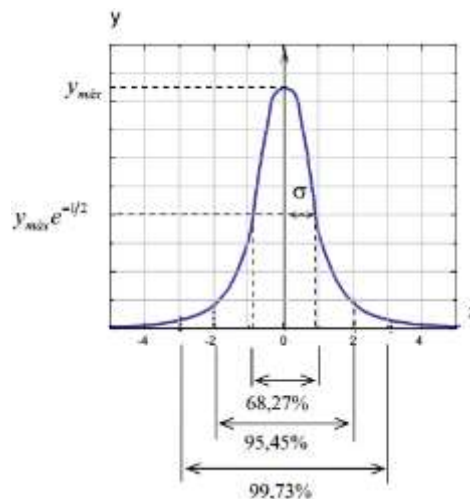
### 3.1.3 Incerteza de Medição

O VIM ainda descreve a incerteza de medição como um parâmetro associado ao resultado de uma medição, que caracteriza a dispersão dos valores que podem ser fundamentalmente atribuídos a um mensurando. A incerteza de uma medição não é outra coisa senão a sua qualidade. Um resultado sem a representação da incerteza associada é um resultado vago e, portanto, não confiável.

Uma medição é representada por um intervalo no qual se espera que o valor verdadeiro esteja contido, bem como um nível de confiança que expressa a credibilidade dessa medição em termos de probabilidade. Assim, o resultado de uma medição é dado por  $\bar{x} \pm k\sigma$ , onde  $\bar{x}$  é uma estimativa do valor esperado, e  $k\sigma$  a estimativa do erro associado à medição, conforme se verifica na Figura 2. O valor  $k$  indica o nível de confiança, sendo 68,27% para a incerteza padrão ( $k = 1$ ), 95,45% para dois desvios padrões ( $k = 2$ ), e 99,73% para três desvios ( $k = 3$ ), para o caso de uma distribuição normal, como ilustrado na Figura 3.



**Figura 2** - Representação de uma medição.



**Figura 3** - Distribuição normal.

Existem duas categorias de incertezas, ou seja, incertezas do tipo A e incertezas do tipo B. As incertezas do tipo A são avaliadas através da análise das flutuações estatísticas, e as incertezas do tipo B através de outros métodos que não sejam a análise estatística de séries de observações.

### 3.1.3.1 Avaliação da Incerteza tipo A

É um método probabilístico de determinação de incertezas, que tem como hipótese fundamental que uma série de medidas diretas de uma grandeza física experimental aproxima-se da Distribuição Normal ou Gaussiana, à medida que se aumenta o número de observações. Sendo  $n$  o número de observações obtidas, o resultado é dado pelo valor médio das medições:

$$\bar{x} = \frac{x_1+x_2+\dots+x_n}{n} = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

Utiliza-se o conceito de desvio padrão ( $S$ ) para avaliar o quanto o valor médio difere do valor verdadeiro ou real da grandeza. Para um número de observações finito, o desvio padrão é chamado de desvio padrão amostral, e é dado por:

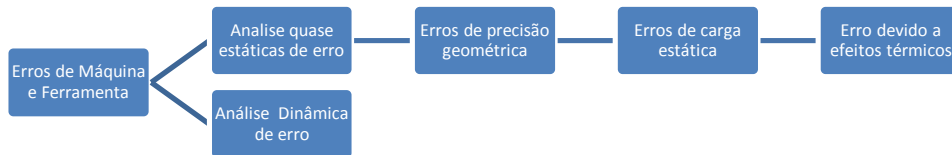
$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \quad (2)$$

### 3.1.3.2 Avaliação da Incerteza tipo B

Essa classificação é dada para resultados de medição obtidos a partir de informações sem caracterização estatística necessária, tais como medições prévias, experiência ou conhecimento geral do comportamento e propriedades de matérias e instrumentos relevantes, especificações do fabricante, relatórios de calibração, manuais e outras especificações.

### 3.2 Hierarquia de erros em máquinas

Os erros identificados podem ser hierarquizados como está mostrado na Figura 4 a seguir. Entretanto, não são apresentados os desvios referentes à análise dinâmica, pelo fato desta não ser contemplada neste trabalho.



**Figura 4** - Hierarquia de erros em máquinas. Fonte: autor.

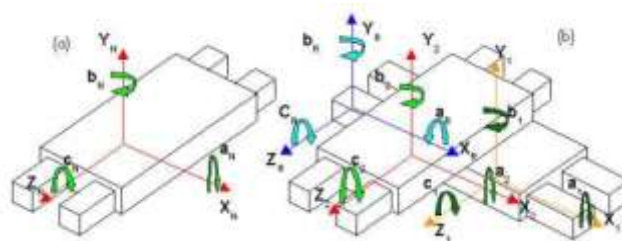
### 3.3 Precisão geométrica da máquina

A respeito do processo de avaliação pode-se citar a norma ISO230–1, que define os procedimentos de teste para avaliação de tolerância. Porém esses testes não refletem no desempenho final da peça, já que, segundo a norma, este processo de medição deve ser realizado com a máquina operando sem carga.

Segundo (Stoeterau, 2004) a qualificação de uma máquina-ferramenta deve ser realizada tomando por base os resultados obtidos com:

- Análise de perpendicularismo;
- Linearidade;
- Posicionamento.

Portanto a análises de resultados apresentadas neste trabalho qualifica a máquina nestes três tipos de análises mencionadas. A Figura 5 representa os graus de liberdade presentes na máquina, de onde podem surgir erros associados.



**Figura 5** – Erros geométricos em guias de máquinas (STOETERAU)

#### 3.3.1 Ensaios geométricos

Segundo (Stoeterau, 2004), de forma geral tem-se constatado que, em centros de usinagem de pequeno e médio porte os erros geométricos são normalmente de posicionamento linear, histerese e de perpendicularismo entre eixos. Os erros de retilinearidade em máquinas novas geralmente apresentam valores muito pequenos, os quais tendem a se tornar mais expressivos com o passar do tempo. Isto se deve a problemas operacionais tais como deficiência de lubrificação, colisões, carregamento excessivo e operação em uma única região de trabalho. Ainda conforme relata Stoeterau (2004), os erros de inclinação, em condições normais de operação, não constituem um grande problema devido às pequenas dimensões relativas do volume de trabalho. A não disponibilidade de

uma rotina de correção bidirecional no CNC implementado no protótipo é um dos fatores que leva à adoção de ensaios unidirecionais. Apesar das considerações dos especialistas da empresa Maho serem para centros de usinagem (MAHO, 1990), essas considerações também podem ser estendidas para os demais tipos de máquinas-ferramentas, de onde se conclui que:

- Ensaios de posicionamento linear, histerese e perpendicularismo podem ser suficientes para prover informações sobre as considerações gerais do comportamento geométrico da máquina.
- E que os ensaios de retilidade são importantes.

Segundo Sousa (2000), um método para medir incertezas relacionadas a retilidade e perpendicularismo é fazer uma análise utilizando uma placa de furos. Os tópicos a seguir apresentam os métodos utilizados para a análise de retilidade e perpendicularismo.

### 3.3.2 Ensaios de Repetitividade de Posicionamento

Os ensaios de precisão e repetitividade de posicionamento devem ser realizados de acordo com a norma ISO 230-2 (1988). Sendo que todas as recomendações desta devem ser seguidas. Durante os ensaios devem ser coletados os valores que serão utilizados pela rotina de compensação, caso haja uma. Cada eixo coordenado da máquina deve ser avaliado de forma unidirecional, com as aproximações sendo realizadas em ambas as direções do sentido do eixo, de forma sincronizada.

### 3.3.3 Ensaios de Perpendicularidade

O desvio de perpendicularidade entre os eixos X e Y é determinado a partir da variação das posições X medidas em relação às referências, ao longo do eixo Y. Para isso, é realizada a comparação entre posições X de referência e as medidas, obtendo-se a matriz M:

$$P = X_{ref} - X_{med} \quad (3)$$

Após são calculadas as melhores retas horizontais que se ajustam aos pontos ao longo de cada linha da matriz, de acordo com a equação:

$$Reta = A x + B \quad (4)$$

Sendo A e B são dados por:

$$A = \frac{n \sum x_i y_i - \sum y_i x_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (5)$$

$$B = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (6)$$

Onde,  $x_i$  e  $y_i$  são as coordenadas dos pontos por onde se ajusta a reta, e n, o número de pontos utilizados no ajuste. O ângulo de inclinação de cada reta é dado por:

$$\alpha_i = \arctg(m_i) \quad (7)$$

O desvio de perpendicularidade é calculado através da média dos ângulos de inclinação das retas ao longo da placa de acordo com a Equação (2).

### 3.3.4 Ensaio de Retilidade

O desvio de retilidade é obtido a partir das variações nas posições dos furos na direção y, quando se desloca ao longo da placa, na direção x. Calcula-se a diferença entre as posições Y de referência e as medidas, segundo:

$$R = Y_{ref} - Y_{med} \quad (8)$$

Por sua vez,  $Y_{ref}$  é calculado utilizando o método dos mínimos quadrados, e representa a reta que mais se aproxima de todos os pontos, neste intervalo. A equação para determinação dessa reta é:

$$Y_{ref} = A x + B \quad (9)$$

Onde A e B são dados pelas equações (5) e (6). E o desvio padrão pode ser calculado pela Eq (2) a partir da matriz R.

## 4 Metodologia

Como ponto de partida para o trabalho de implementação da máquina proposta, os seguintes critérios de projeto foram definidos: utilização de componentes de baixo custo, mas que não comprometam as tolerâncias estabelecidas para o projeto; um *software* com interface amigável e robusto, capaz de efetuar automaticamente a calibração e correção de desalinhamento do material a ser usinado. Uma visualização virtual completa em três dimensões, onde o usuário poderá editar e adicionar rotinas de código para a execução da máquina. Necessita-se, também, implementar uma solução que possa diminuir possíveis erros críticos, como o escorregamento de passos nos atuadores. Para uma melhor estruturação do projeto esse capítulo foi subdividido em três etapas, conforme segue:

- **PROJETO MECÂNICO DA MÁQUINA**

Essa etapa envolve a elaboração do desenho da máquina, realização de simulações de deformação, identificação e seleção dos componentes utilizados, tais como rolamentos lineares, guias lineares e fuso de esferas, assim como suas características de tolerância, em contrapartida aos custos.

- **PROJETO ELÉTRICO (ATUADORES E *DRIVERS* DE POTÊNCIA)**

O projeto da parte elétrica apresenta a definição dos *drivers* de potência para acionamento dos motores, a definição das fontes de alimentação que serão utilizadas, e o cálculo da potência necessária dos atuadores para a escolha dos mesmos, levando-se em consideração o custo.

- **PROJETO DO *SOFTWARE*, *HARDWARE* E PROTOCOLOS DE COMUNICAÇÃO**

Define a implementação do *software* de controle da máquina. Nesse sentido, deverão ser levadas em consideração todas as funções que visam facilitar o uso da máquina, de forma que “qualquer usuário” possa utilizar a mesma sem despendar de muito tempo para aprender a operá-la. Desta forma, a respeito deste tópico, deverão ser definidos os fluxogramas de operação do *software*, assim como do protocolo de comunicação. Também deverá ser escolhida a plataforma para implementação do mesmo.

### 4.1 Projeto Mecânico da Máquina

O projeto mecânico foi desenvolvido com o auxílio da ferramenta de desenho *Solidworks*. Porém, antes de realizar o desenho do projeto em si, devem-se estabelecer alguns pré-requisitos, os quais podem ser definidos como:

- Máxima deformação elástica;
- Tolerância a folgas;
- Custos;
- Processo de fabricação;
- *Design* e acabamento.

Com base nos fatores que foram levados em consideração no projeto, e pelo fato deste ser apenas um protótipo, tratou-se com maior prioridade os fatores de custo e processo de fabricação. Para atender a esses fatores, realizou-se uma pesquisa criteriosa a respeito dos processos de fabricação que pudessem ser terceirizados, levando-se em consideração o custo como fator determinante. Através desta, a solução encontrada para a construção da estrutura principal da máquina foi a utilização do processo de recorte com plasma e dobra de chapas de aço, com posterior união por meio de soldagem MIG. Este processo de soldagem foi escolhido porque, primeiramente, dos processos disponíveis para este projeto verificou-se três tipos, sendo eles: TIG, solda a eletrodo, e solda MIG. Neste sentido escolheu-se solda MIG por esta conferir melhores resultados em relação a acabamento e propriedades do processo de solda.

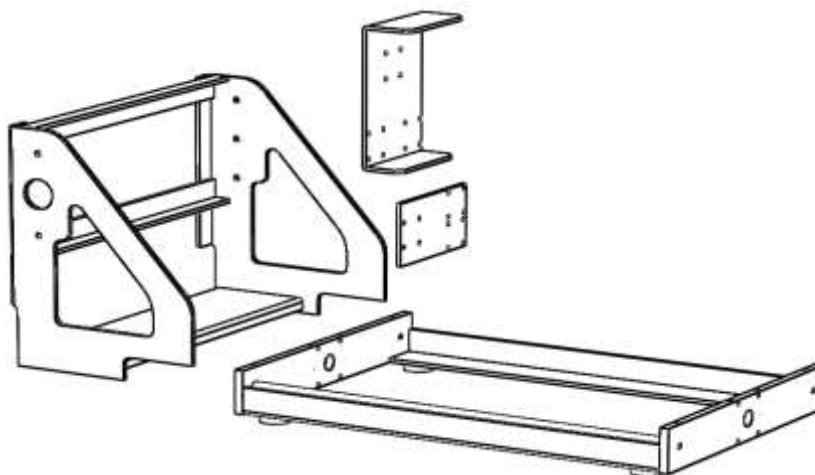
Considerando os processos de fabricação já pré-estabelecidos, exigiu-se a definição das características específicas da máquina, como por exemplo:

- Se a mesa de fixação será móvel ou fixa;
- Quais guias serão utilizadas;
- Qual será o sistema de movimentação (fuso ou correia);
- O comprimento e largura da mesa;
- Máximo deslocamento longitudinal da mesa;
- Máximo deslocamento transversal;
- Máximo deslocamento vertical do suporte da mesa;
- Máxima altura da superfície da mesa em relação ao eixo principal;
- Peso que a máquina suporta sobre a mesa.

Sendo assim, para obter-se uma forma ideal, foram elaborados aproximadamente 20 esboços. A grande dificuldade de elaborar a forma da estrutura deve-se a uma combinação de fatores como: desenvolver as formas com base nos recursos disponíveis, aliar o design e acabamento da máquina com os requisitos mecânicos de tolerância e resistência. Esse processo pode parecer mais simples do que realmente é, porém, na prática, percebeu-se que esta foi uma das partes mais trabalhosas no processo de criação do protótipo. Como mencionado, alguns esboços foram elaborados, e a cada esboço criado, foi possível identificar novos erros e, então, corrigi-los. Porém, apesar de serem realizados inúmeros redesenhos, ainda ocorreram algumas falhas, as quais só foram identificadas após a construção do protótipo, sendo que ao todo foram realizados três reprojatos na parte estrutural da máquina.

Em síntese, o motivo dos retrabalhos efetuados foi o mal posicionamento dos furos, desalinhamentos, entre outros. A Figura 6 apresenta o resultado final do projeto da forma da máquina cujas dimensões gerais são 250 mm x 360 mm x 550 mm.





**Figura 6** - Vista explodida da fresadora. Fonte: Autor.

A estrutura foi projetada com recortes de chapas, portanto, para executar cada recorte, foi necessário criar um desenho no formato DXF, padronizado para máquinas de corte a plasma. Esses desenhos estão mais detalhados no Anexo I.

Antes de passar para a execução do projeto, foi verificado se o modelo era viável e se atendia aos fatores de projeto. Nas sessões subsequentes, é explicado como foi feito e os resultados obtidos pelas simulações, bem como a seleção dos componentes mecânicos.

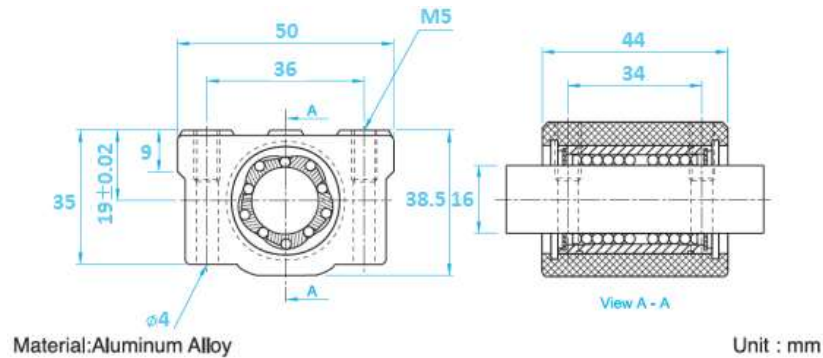
#### 4.1.1 Seleção dos Componentes Mecânicos

Este tópico tem por objetivo detalhar quais os componentes mecânicos que foram utilizados no projeto dentre as diversas opções disponíveis no mercado, bem como o porquê de terem sido escolhidos. Esses componentes estão listados a seguir:

##### 4.1.1.1 Rolamentos Lineares

A fim de conferir um menor erro, relacionado às incertezas de alinhamento e de deslocamento, foram utilizados rolamentos lineares inseridos em mancais comercialmente chamados de *Pillow Blocks*. Estes mancais proporcionam um auto alinhamento durante a montagem, já que são fixados lateralmente por quatro parafusos.

Esta forma de separar o sistema em partes independentes, fixadas na montagem através de componentes de fixação, como parafusos, rebites, e outros, pôde assegurar resultados mais satisfatórios para a construção da máquina, uma vez que sua construção foi baseada apenas em ferramentas de baixo custo. Como alternativa a este processo poderia ter sido utilizado o processo de fundição ou usinagem de uma estrutura totalmente alinhada, porém, estes elevariam o custo total do projeto. Na Figura 5 está ilustrado o rolamento utilizado na máquina.



**Figura 7** - Rolamento linear *Pillow Block* – SMA16-UU – fonte: Kalatec, 2013a.

#### 4.1.1.2 Guias Lineares

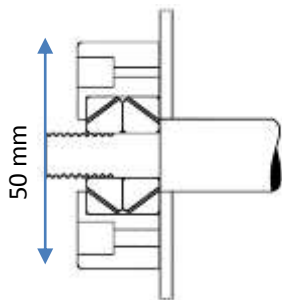
As guias de eixo utilizadas na construção do protótipo proporcionaram ao sistema menores distorções de alinhamento, restringindo as tolerâncias mecânicas da máquina. As guias lineares devem ser utilizadas juntamente com os rolamentos lineares. O constante contato entre o rolamento linear e a guia pode causar desgastes com o tempo. Para retardar este fenômeno é indicado o uso de tratamentos térmicos superficiais nas guias. A empresa Kalatec Automação, fabricante das guias utilizadas, menciona em seu catálogo ter utilizado o tratamento de têmpera por indução, além de outras informações como dureza superficial na faixa de SUJ-2 / S55C, e tolerância diametral h6 (Kalatec, 2013b).

#### 4.1.1.3 Fuso de Esferas

O fuso de esferas é um sistema mecânico que transfere movimento rotacional para movimento linear. O sistema é composto por um eixo rosqueado e um invólucro chamado de castanha, por onde as esferas circulam com baixo atrito entre esses dois componentes. A castanha possui um sistema de retorno para as esferas voltarem ao início do caminho. O fuso é fixado em mancais em suas extremidades e recebe o movimento de rotação do motor, provocando o movimento linear da castanha, a qual é presa à carga a ser movimentada.

#### 4.1.1.4 Rolamentos e mancais

Para a seleção dos rolamentos, levaram-se em consideração dois aspectos necessários: um deles é o dimensionamento, de modo que eles pudessem ser acoplados ao protótipo; o outro, e também o aspecto mais importante, é obter um mancal que “elimine” todos os graus de liberdade, com exceção da rotação do eixo. Outros aspectos como vida útil e resistência mecânica foram classificados como fatores secundários. Na Figura 8 a seguir é apresentado um esboço da configuração de rolamentos e mancal onde os aspectos mencionados são satisfeitos.



**Figura 8** - Mancal para rolamentos de fixação do fuso. Fonte: Autor.

#### 4.1.1.5 Elementos de fixação

Como elemento de fixação utilizou-se parafusos do tipo Allen, pois em geral possuem uma alta resistência em relação à carga aplicada, bem como um melhor acabamento. Estes parafusos são classificados segundo a norma ABNT NBR ISO 724:2004 como M5, M8.



**Figura 9** - Parafusos utilizados na fixação das partes mecânicas da fresadora CNC. Fonte Promaq, 2013.

Os elementos de máquinas foram conferidos segundo a funcionalidade no desenho em CAD. Além deste motivo, estes elementos tiveram a necessidade de atenderem aos fatores de projeto tais como custo, tolerância e *design*. Cabe ressaltar, ainda, que para este trabalho não foi realizado um estudo mais aprofundado relacionado à resistência dos materiais, tendo em vista que seria inviável se aprofundar em um tópico em específico, haja vista a diversidade de conteúdo a ser apresentado.

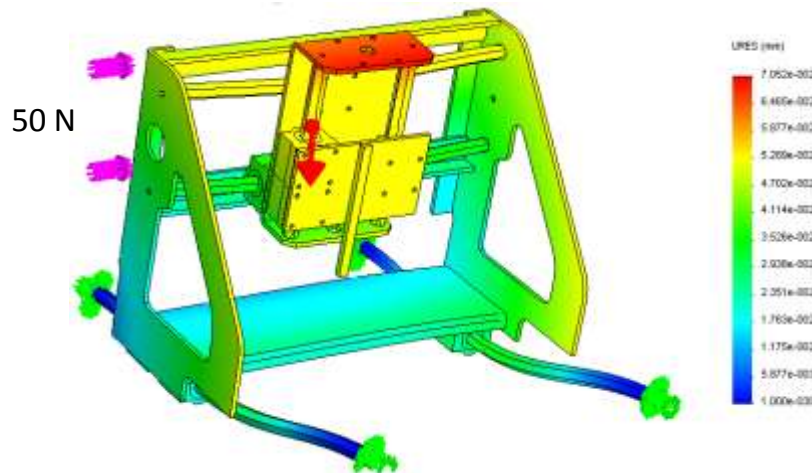
#### 4.1.2 Simulação e verificação do modelo

Para avaliar as deformações utilizou-se uma ferramenta do *software Solidworks* chamada *Solid-Simulation*, com a qual foi possível realizar um estudo estático em elementos finitos.

Cabe ressaltar que a utilização desta ferramenta deu-se apenas para ter-se uma noção global de como o sistema iria se comportar com a aplicação de uma carga lateral de 50 N, bem como o próprio peso da estrutura. Conforme se verifica na Figura 10, a estrutura se deforma em uma escala de ampliação de aproximadamente 800 vezes. Como mencionado anteriormente, a ferramenta foi utilizada apenas para se ter uma noção do comportamento estático do sistema mecânico, bem como para auxiliar no desenvolvimento do protótipo.

Além disso, os resultados mostrados na Figura 10 podem não ser satisfatoriamente realísticos, logo, para que realmente se possa confiar nos resultados obtidos pela simulação, deveriam ser realizados estudos e testes que pudessem comprovar ou não sua confiabilidade e eficácia. Para chegar-se ao resultado da figura a seguir, utilizando o *software* mencionado, definiu-se algumas configurações como número de nós, que foi de 14.672, número de

elementos a serem calculados que para este caso foi 47.692, bem como definir as cargas envolvidas assim como esta exemplificado na figura:



**Figura 10** – Estudo de deformação obtido pelo *Solid-Simulation*. Fonte: Autor.

Pode se perceber pela Figura 10 que existe uma grande deformação nas guias lineares, porém, esta deformação é proveniente do peso da estrutura. Este erro pode parecer um grande problema, que de fato é, porém, como se trata de um erro sistemático, afetando apenas a exatidão, este poderá ser corrigido por *software*.

## 4.2 Projeto Elétrico

A respeito do projeto elétrico, pode-se dividi-lo em três partes: inicialmente, são explicados quais os critérios utilizados para a seleção dos atuadores; na sequência, é feita uma explicação de como foi realizada a seleção e escolha dos *drivers* de potência para acionamento dos motores; por fim, é mostrado como foi realizada a seleção da fonte de alimentação, bem como foi construído o painel elétrico, que nada mais é do que o espaço físico no qual os componentes elétricos estão agrupados.

### 4.2.1 Seleção dos atuadores

Para a escolha dos motores é importante saber qual a força de avanço estabelecida no projeto. Considerando que essa força é igual a 50 N para uma velocidade de avanço de 1000 mm/min, e sabendo-se que a potência é dada pelo produto da força pela velocidade de avanço em metros por segundo, verificou-se que a potência máxima de avanço da máquina para as condições citadas é dada por:

$$P_a = V_a F_a = 5 \text{ W} \quad (10)$$

Conforme se verifica na curva do torque da Figura 11, o fator crítico para se determinar as características do motor não se deve ao cálculo da potência, mas sim à curva do torque, pois para motores de passo essa curva decai rapidamente com o aumento da velocidade. Já a estimativa da força de atrito permanece a mesma, podendo até mesmo aumentar, caso seja considerado o atrito viscoso. Assim, considerando que o fuso tem um passo de 5 mm por revolução, a velocidade angular estabelecida é de  $W_n = 22 \text{ rad/s}$  (210 rpm).

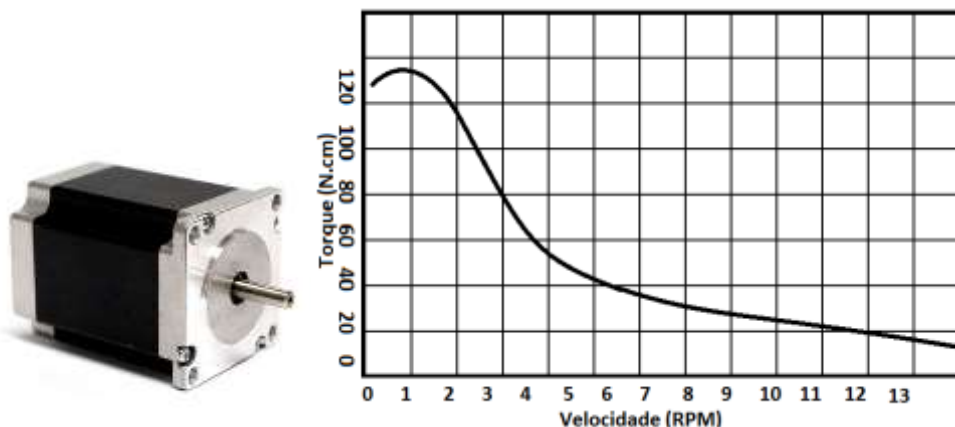
Para obter-se o resultado do torque causado pelo atrito, foram realizados testes práticos, chegando-se ao valor de  $T_{atr} \approx 0,5 \text{ Nm}$ . Porém, esse valor pode variar de acordo

com a força de avanço, sendo que o torque medido é validado para as condições citadas. Sendo assim, o somatório dos torques nestas condições pode ser expresso por:

$$\sum T = \frac{P_a}{W_n} = T_{atuador} - T_{atr} \quad (11)$$

$$T_{atuador} = 50 \text{ Ncm}$$

Para que a máquina se movimente nas condições estabelecidas é necessário que o motor forneça um torque igual ou superior ao torque calculado pela Eq. (11)



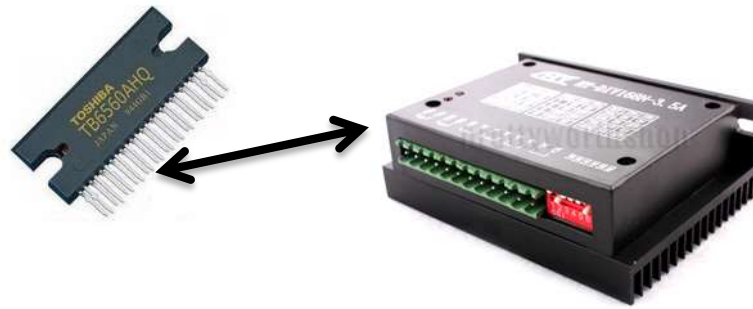
**Figura 10** - Motor de passo e curva de torque. Figura redesenhada com base na figura do fabricante Sanyo.

O motor escolhido possui *holding torque* de 110 N.cm, e torque de trabalho de aproximadamente 50 N.cm na velocidade nominal da máquina, definida no projeto, como pode ser conferido na curva de torque fornecida pelo fabricante.

#### 4.2.2 Seleção dos Drivers de potência

O *driver* de potência selecionado foi o TB6560AHQ, cujas características podem ser visualizadas no Anexo 1. Este *driver* foi escolhido por alguns fatores: primeiramente, porque dentre os *drivers* de motores de passo este era o mais viável economicamente. E também por haver grande disponibilidade de circuitos prontos que o utilizassem. Na Figura 12 a seguir é apresentado o circuito integrado e uma das alternativas de encapsulamento vendidas por fabricantes chineses, que consiste basicamente em uma caixa com dissipador e alguns periféricos de segurança, para fornecer a tensão de alimentação necessária para os motores. Os *drivers* de potência, por sua vez, devem também conter um dispositivo opto-acoplador, que serve para isolar o circuito de potência e conter um possível refluxo de corrente proveniente das bobinas do motor de passo.

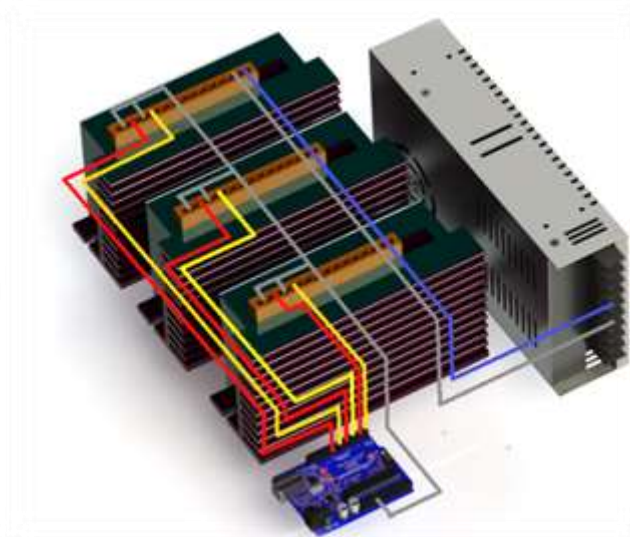
O circuito integrado TB6560AHQ, que é fabricado pela Toshiba, possui alimentação de 12 V e pode controlar motores de passo bipolares. Ele fornece até 3,5 A de corrente por fase do motor, e pode interpolar a tensão das fases, aumentando a resolução do motor em até 16 vezes. Considerando que a resolução nominal dos motores é de 200 passos por volta, se estes forem acionados pelo *driver*, essa resolução passa a ser de 3.200 passos por revolução.



**Figura 11** – Driver de potência TB6560AHQ. Construção do painel elétrico e seleção da fonte de alimentação.

Após a seleção dos *drivers* de controle dos motores, fez-se necessária a escolha de uma fonte de alimentação a qual pudesse suprir a potência necessária para a operação do sistema. A potência dissipada pelos *drivers*, segundo o catálogo do fabricante, é em torno de 15 watts, então, a potência total gasta pelo sistema foi cerca de 300 watts. Sendo assim, optou-se pela utilização de uma fonte de 350 W, com tensão variável de 30 a 36 V e corrente máxima de 10 A.

Foi feito um módulo contendo os drivers, a fonte, e o microcontrolador Arduino Uno R3, como ilustrado na Figura 12. As ligações em vermelho representam os sinais de avanço de passo, enquanto os sinais em amarelo representam o sentido de giro do motor. As ligações em cinza e azul representam o terra (GND - lógico e de potência) e tensão de corrente contínua (VCC), respectivamente. Na Figura 13 é apresentado um diagrama com as conexões utilizadas entre o Arduino e os drivers dos motores.



**Figura 12** – Esquemático do sistema de hardware de controle CNC.

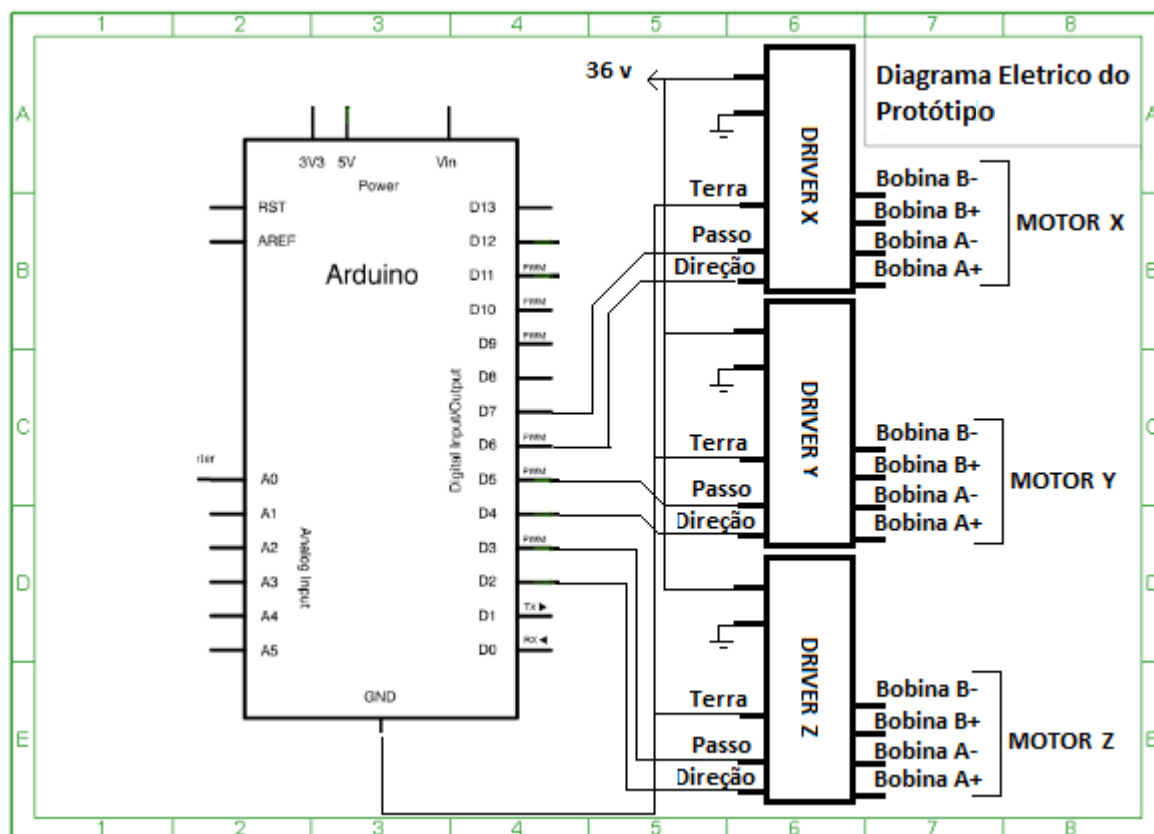


Figura 13 – Diagrama de conexões entre o controlador e os drivers.

### 4.3 Projeto do software, hardware e protocolo de comunicação

O *software* de controle é responsável por tarefas como o interfaceamento entre o usuário e a máquina. São diversos os tipos de *softwares* disponíveis no mercado, porém, nenhum deles mostrou-se satisfatório para a operação da máquina construída devido a fatores como interface gráfica pouco intuitiva, atualização para novas plataformas (Windows 7, Windows 8, etc), necessidade de utilização de porta paralela, impossibilidade de implementação de controle em malha fechada. Além disso, nenhuma destas ferramentas são acessíveis gratuitamente.

Portanto, fez-se necessária a criação de um *software* específico, cujas características suprissem os problemas mencionados anteriormente.

#### 4.3.1 Projeto do software

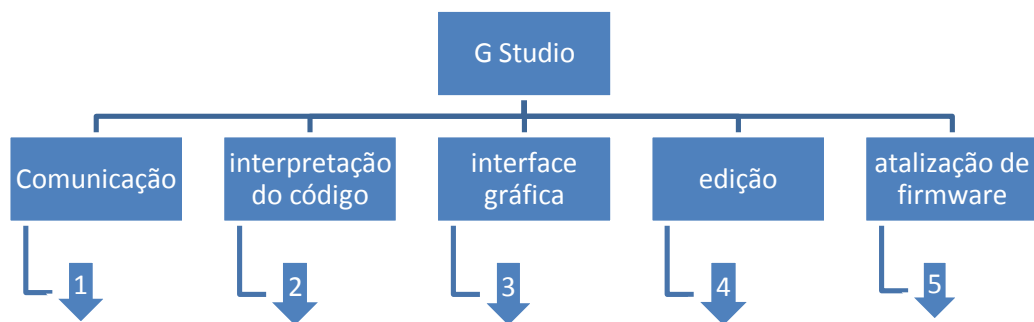
Inicialmente é importante mencionar que o objetivo geral do *software* é a implementação da interface entre o operador da máquina e o usuário. Esta tarefa, por assim dizer, parece ser mais simples do que realmente se constatou. Até o momento, no *software* já foram utilizadas cerca de 6 mil linhas de código, apesar de ser construído todo em linguagem C++ com o auxílio do editor visual. O uso da linguagem C++ facilita bastante no

desenvolvimento, já que esta é uma linguagem de programação orientada a objetos e existem muitas funções prontas e/ou facilmente encontradas.

Para se implementar esta interface, definiram-se algumas necessidades importantes, conforme se verifica a seguir, sendo em seguida apresentadas as soluções.

- Funcionamento em plataformas atuais;
- Comunicação USB;
- Facilidade para executar o código;
- Visualização do desenho em três dimensões;
- Posicionamento do código, bem como definição de escala e rotação em torno de cada eixo (X, Y e Z);
- Sistema de atualização do *firmware*;
- Sistema de configuração manual da máquina;
- Edição gráfica do código;
- Edição textual do código.

Para atender a estes propósitos, definiu-se previamente o fluxograma do *software* de controle especificamente para este trabalho, denominado “Gstudio”. A Figura 14, apresenta-se o fluxograma principal do programa.



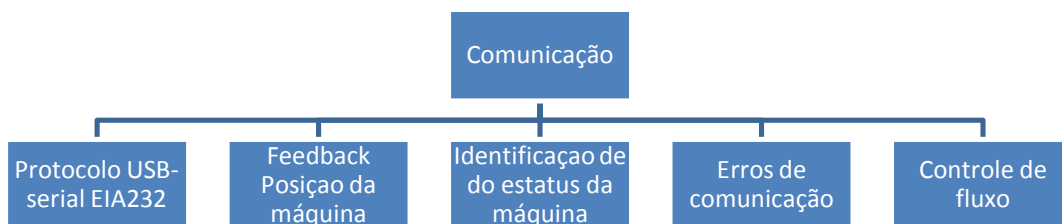
**Figura 14** –Hierarquia principal do software de interface (Gstudio).

Depois de definida a estrutura do problema, foram especificadas características para cada ponto. Sendo assim, foram criados fluxogramas correspondentes a cada tópico, sendo na sequência comentada a implementação de cada um destes.

#### 4.3.1.1 Comunicação

A comunicação entre a interface gráfica (Gstudio) e o *firmware* do controlador (Grbl) pode ser subdividida em alguns tópicos, assim como mostrado no fluxograma da Figura 15 a seguir:





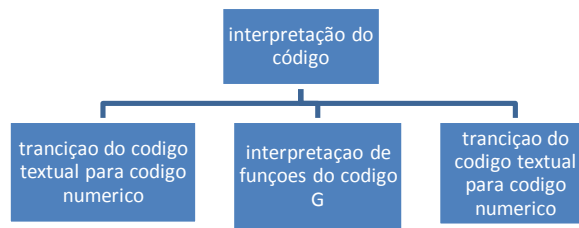
**Figura 15** – Hierarquia de tópicos referente à comunicação entre Gstudio e Grbl.

- **Protocolo USB-Serial EIA232:** Utilização do protocolo USB serial, este já está implementado na comunicação da plataforma Arduino, podendo ser facilmente alterado.
- **Feedback de Posição da Máquina:** O software está baseado no sistema de comunicação do “Grbl”, que será explicado na seção subsequente. Desta forma, para que o programa saiba a posição da máquina, é necessário “perguntar ao hardware” em qual posição a máquina está. Esta solicitação pode ser enviada a qualquer momento de forma assíncrona, porém como o sistema do controlador tem velocidade de processamento limitado, a taxa de para este *feedback* deve se limitar a 26 Hz segundo Simen, 2012, que é o autor do Grbl.
- **Identificação de Status da Máquina:** Identificar se há a presença de alarmes ou se a conexão está *online* ou ainda se está *off-line*.
- **Erros de Comunicação:** Se por algum motivo a comunicação cair o status da máquina mudar para *off-line* o software passará para o modo de busca que varrerá todas as portas virtuais e reais até encontrar a porta correta e se reconectar.

**Controle de fluxo:** O controle de fluxo é responsável pela parte mais importante da comunicação, que é identificar qual o momento certo para se passar os comandos ao hardware sendo que este tem um buffer de armazenamento de comandos. Se um comando é enviado e o buffer não está cheio o interpretador envia uma resposta “ok”. Porém quando um comando é enviado e o *buffer* está cheio nenhuma resposta é recebida até que libere um espaço no *buffer*. Sendo assim quando um comando é enviado sem que o comando anterior receba uma resposta de “ok” então este novo comando é descartado, desta forma todo comando apenas pode ser enviado do “Gstudio” para o “Grbl” depois da resposta do Grbl para o comando anterior.

#### 4.3.1.2 Interpretação do código G

A parte do Gstudio destinada à interpretação do código G pode melhor compreendida dividindo-a em três tópicos, como está colocado a seguir, e sendo explicado cada um deles na sequência.

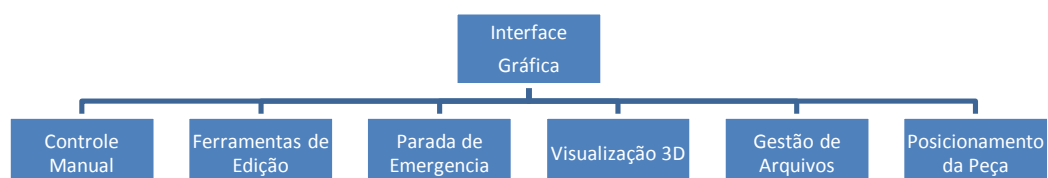


**Figura 16** – Hierarquia de processos de interpretação de código G. Fonte: Autor.

- **Transcrição do Código Textual para Código Numérico:** É a parte do código responsável por transformar o código em formato de texto para formato numérico, desta forma criou-se uma matriz numérica de duas dimensões onde uma dimensão é o número da linha do código e a outra são os parâmetros e tipo do código numa sequência fixa expressa da seguinte maneira:
- **Interpretação de Funções do Código G:** O Software realiza interpretações das funções antes de agrupadas na matriz funções como G02, G03, G51, G90 entre outras que precisam ser processadas antes de serem encaminhadas para o controlador.
- **Transcrição do Código Textual para Código Numérico:** Esta tarefa executa a interpretação numérica da matriz para linhas de comandos textuais que posteriormente serão enviadas para o hardware de controle da máquina.

#### 4.3.1.3 Interface gráfica

A interface gráfica propriamente dita é a parte do *software* onde se agregou maior importância devido aos objetivos que foram estabelecidos, onde deverá se priorizar a facilidade de utilização pelo usuário. Para se identificar quais os pontos trabalhados no quesito interface criou-se o diagrama apresentado na Figura 17.



**Figura 17** – Fluxograma da interface do Gstudio. Fonte: Autor.

- **Controle Manual:** O controle manual pode ser executado de duas formas, uma delas é através do mouse, onde o usuário deverá clicar no espaço 3D da interface gráfica. A outra forma de controlar a máquina é pelo teclado. Além destas formas

de controle a interface possui controles na barra de ferramentas onde o usuário pode ativar propriedades como velocidade e aceleração de avanço para o controle manual bem como níveis de velocidade do mandril (*spindle*).

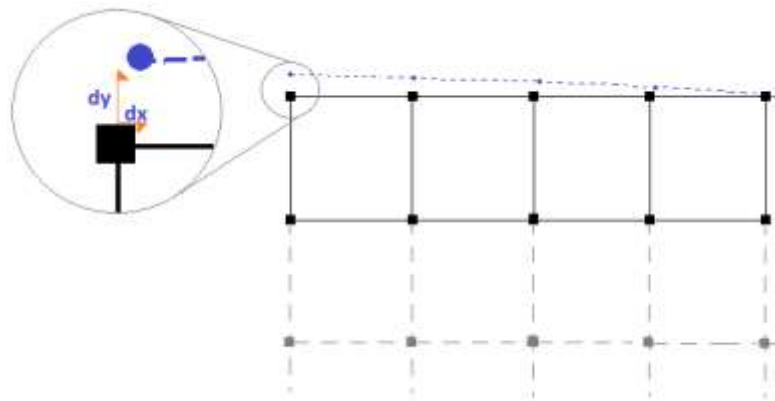
- **Ferramentas de Edição:** Para que o usuário possa editar o código de forma mais intuitiva foi criado um menu de ferramentas CAD onde usuário pode inserir formas como: linha, arco, círculo, retângulo, polígonos e etc. O código G em sua forma textual é automaticamente atualizado de acordo com as alterações gráficas executadas.
- **Parada de Emergência:** A máquina possui três formas de parada pelos botões gráficos de “*stop*”, “*pause*” e “*e-stop*” sendo que este último, possui um atalho no teclado (Alt+Enter). O botão stop tem a função de parar a máquina e interromper o ciclo em operação, o botão pause tem a função de parar a máquina, porém não cancela o ciclo em operação, de forma que quando o usuário reiniciar a execução, máquina retornara na situação onde tinha parado. E por último o botão *e-stop* é um botão de emergência assim como especificado pela norma EN-292-2, ele atua de forma mais simples possível desligando a alimentação parte de potência, porém ainda é um comando por software sendo assim ainda é importante ter um botão físico no painel para desligamento seguro.
- **Visualização 3D:** Desenvolvida pela biblioteca gráfica *OpenGL* visualização 3D tem a função de reproduzir a trajetória exata da ferramenta e também tem o objetivo de mostrar previamente qual será a trajetória que será executada. A visualização tem funções de *pan* para desloque a câmera no plano de visualização, também tem a função de rotação e *zoom* que pode ser executada pelo mouse ou pelo teclado.
- **Gestão de Arquivos:** Tem a função de abrir importar e exportar arquivos para as diversas extensões existentes de código G.
- **Posicionamento da Peça:** Estas ferramentas possibilita o posicionamento do código G por completo, isto pode ser feito de duas maneira, pelas ferramentas “*Move*”, “*Scale*” e “*Rotate*” onde o usuário irá inserir os respectivos dados manualmente em cada campo, ou pode ser feito utilizando-se da ferramenta “*SmartPosition*” onde o usuário ira movimentar a máquina em dois pontos distintos, e os campos mencionados antes serão atualizados para que o código se enquadre entre os dois pontos fornecidos.

#### 4.3.1.4 Atualização de firmware

A instalação do *firmware* é facilitada através de uma ferramenta que possibilita gravar automaticamente uma imagem do sistema interpretador de comandos no Arduino. O instalador do software Juntamente com a máquina, é disponibilizada esta ferramenta, que permite a atualização do firmware com uma imagem atualizada do sistema Grbl. nte, sem a necessidade de um *software* externo, como por exemplo, a IDE do Arduino.

#### 4.3.1.5 Sistema de Calibração

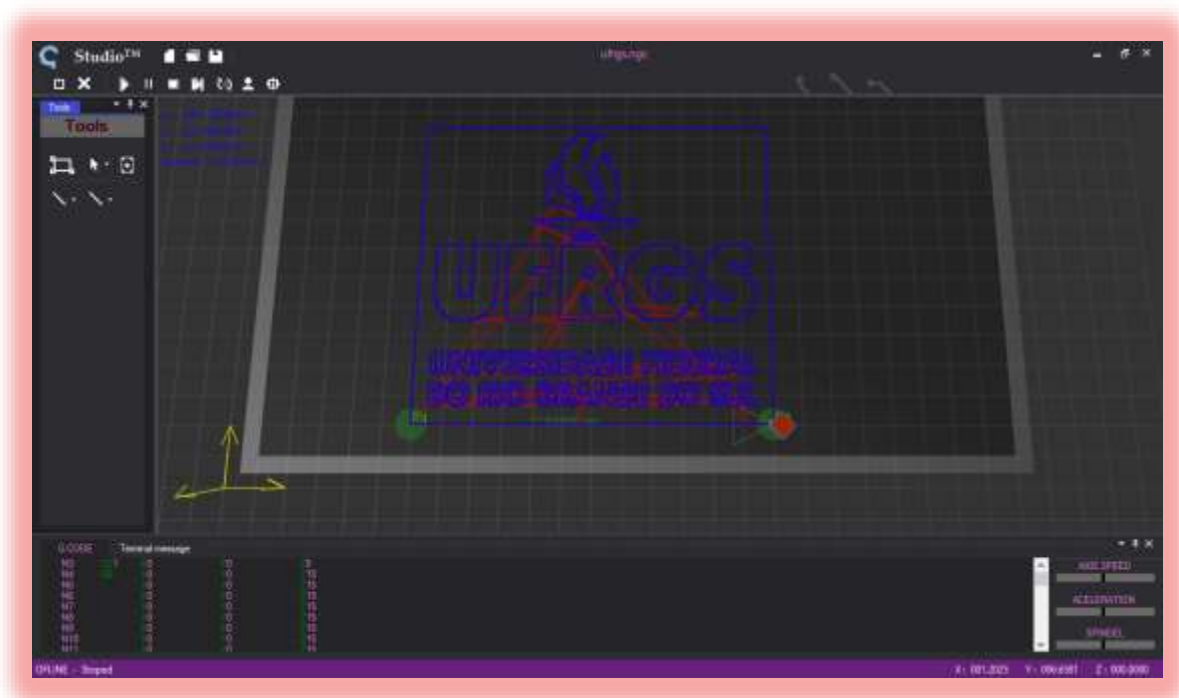
Para corrigir o erro determinístico da máquina, ou erros sistemáticos, criou-se uma função no Gstudio, através da qual o usuário preenche uma matriz de *offset*, cujas dimensões podem ser configuradas. Desta forma, pode-se diminuir o erro de *offset* em relação à posição alvo desejada, porém, não foi implementada no *software* um método que corrija o erro durante a trajetória.



**Figura 18** – Representação da matriz de calibração de duas dimensões no campo de trabalho.

A respeito do sistema descrito na Figura 18, são mostrados alguns pontos marcados na cor preta. Tais pontos representam a suposta posição para a qual a ferramenta deveria ir quando comandada, e o ponto em azul representa a posição média da ferramenta quando comandada para a posição correspondente à matriz teórica. Logo, a função desta ferramenta de calibração é comandar a ferramenta para compensar estes erros  $dx$  e  $dy$ .

A respeito da calibração, ainda é possível calibrar a zona morta existente nos eixos nas figuras. Portanto, a implementação final de todos os itens mencionados pode ser conferida na Figura 19 a seguir.



**Figura 19** – Tela principal do Gstudio.

A é o resultado do *software* criado, sendo que esta é a tela principal, onde estão acessíveis o menu de edição do código, a barra de controle, o campo de visualização e edição textual do código G, além do campo principal, que é a visualização da ferramenta e o desenho criado pelo código G, em um espaço tridimensional (espaço 3D).

#### 4.3.2 Seleção do Hardware

O *hardware* escolhido foi o Arduino (Figura 19) por várias razões, a mais importante foi porque, ele é integrado como o mais popular na comunidade DIY (*do it yourself*). Sendo assim tem-se grande acessibilidade às informações e linhas de desenvolvimentos, além deste ser de baixo custo.



**Figura 20** – Plataforma de controle Arduino Uno R3. Fonte: Electron-Shop.

No entanto, para esta tarefa, este controlador apresenta algumas limitações, como falta de memória e baixa velocidade de processamento. Sendo assim, o *firmware* aplicado tem a necessidade de ser suficientemente simples e otimizado.

#### 4.3.3 Seleção do *firmware* do controlador e protocolo

O *firmware* de controle utilizado foi o interpretador de código G denominado *Grbl* pelo indiano Simen Svale Skogsrud (2012). Atualmente a maioria das impressoras 3D da comunidade DIY de código aberto têm o Grbl em seus núcleos de processamento. O Grbl foi adaptado para uso em centenas de projetos, incluindo cortadores a laser, plasma, máquinas de desenho excêntricas, entre outros, devido ao seu desempenho e simplicidade.

O Grbl é projetado para ser simples. Sendo assim não é uma solução completa para todos os tipos fresagem CNC, mas é um ponto de partida para qualquer um construir uma máquina de 3 eixos tipo cartesiano, fresadora, máquina de corte a laser, impressora 3D, etc. Sua comunicação é feita através da porta serial do Arduino e precisa de um fluxo constante de comandos G, o código então é enviado através de um computador ou qualquer outro meio que possa se comunicar através do protocolo EIA-232.

O funcionamento do Grbl consiste, essencialmente, em dois módulos em execução ao mesmo tempo. O programa principal recebe pela porta serial as linhas de comando (código G), as quais são interpretadas e decompostas em blocos de dados. Cada bloco contém informações de uma trajetória em linha reta, e aceleração e velocidade do movimento. Os blocos são colocados em um *buffer*, tipo FIFO (*First In First Out*), de no máximo 16 blocos (no

caso do Arduino 328p). Se o bloco for aceite é retornada uma mensagem 'ok', e caso o *buffer* esteja cheio, é retornado uma mensagem contendo o código do erro, no formato 'erro: cod'.

Enquanto o programa principal é executado, o programa secundário é chamado várias vezes por interrupção. Ele controla os motores de passo, enviando pulsos de passo e direção para os *drivers* dos motores, a partir das informações contidas na fila de blocos. Os eventos são processados sequencialmente através dos blocos de contidos no *buffer* até que ele esteja vazio.

O programa principal aceitará continuamente novos blocos de código G tão rapidamente os blocos possam ser processados. Se a fila de comandos estiver cheia, um comando de 'erro' é retornado. Isto significa que a interface com o usuário deve sempre aguardar por uma resposta de 'ok' ou 'erro' antes de enviar um novo comando de código G. Além disso, o fluxo de dados deve ser constante e ininterrupto de modo a minimizar a chance da máquina sofrer paradas súbitas.

Quanto às interfaces externas, existem apenas interruptores de limite XYZ, sendo necessário fazer algumas alterações no firmware para implementação de *encoders* na máquina. Outras características, como a pausa e parada, redução de velocidade e ciclos de *homing*, ocorrem em tempo real, ou seja, de forma assíncrona.

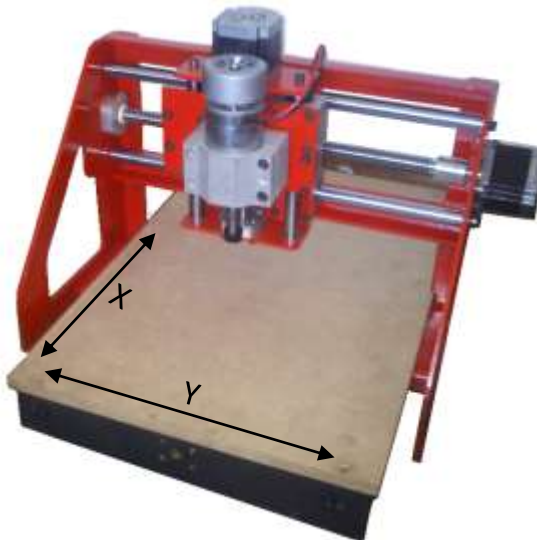
O Grbl implementa um subconjunto do padrão NIST RS274/NG. Os comandos suportados estão na listagem a seguir (Tabela 1).

**Tabela 1** – Comandos suportados pelo interpretador Grbl

Código G	Descrição do comando
G0	Movimento linear rápido
G1	Movimento linear com velocidade controlada
G2	Movimento em arco, sentido horário
G3	Movimento em arco, sentido anti-horário
G4	Pausa o movimento de avanço dos eixos pelo tempo especificado
G17	Seleção do plano X-Y
G18	Seleção do plano X-Z
G19	Seleção do plano Y-Z
G20	Unidades de medida em polegadas
G21	Unidades de medida em milímetros
G28, G30	Retorno para posição de origem
G28.1, G30.1	Seleção de posição predefinida
G53	Outros
G90	
G91	
G92	Compensação de coordenadas
G93, G94	Avanço invertido
M3 / M4	Liga <i>spindle</i> no sentido horário / anti-horário
M5	Desliga o <i>spindle</i>
M0, M2, M30	Programa, pausa e finalização

## 5 Análise dos resultados

Como resultado final da implementação prática apresenta-se o protótipo em seu estágio final, ilustrado na Figura 21.



**Figura 21** – Imagem do protótipo construído.

A fim de analisar os resultados, primeiramente avaliou-se o protótipo segundo sua funcionalidade. Neste sentido, inicialmente foi conferida cada parte do protótipo, incluindo a parte mecânica, observando-se possíveis folgas ou desalinhamentos. Então, posteriormente, testou-se *hardware*, *firmware* e *software*, identificando os sinais de saída. Por fim, realizou-se um novo teste, interligando todas as etapas do projeto e fazendo com que a máquina executasse uma trajetória definida.

### 5.1 Análise de incertezas

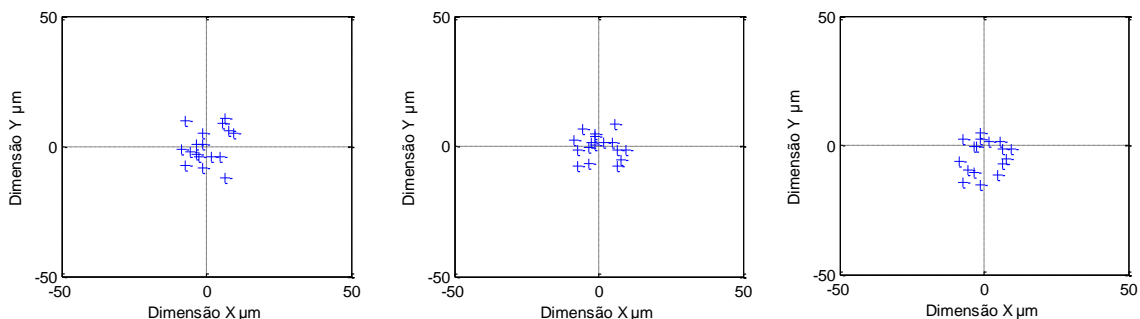
#### 5.1.1 Análise de Repetitividade

Para avaliar a característica de repetitividade da máquina, realizaram-se testes com o auxílio da ferramenta de medição mencionada anteriormente. Esta análise foi realizada em alguns pontos específicos da área de trabalho da máquina, assim como demarcado pelas numerações vistas na Figura 21.

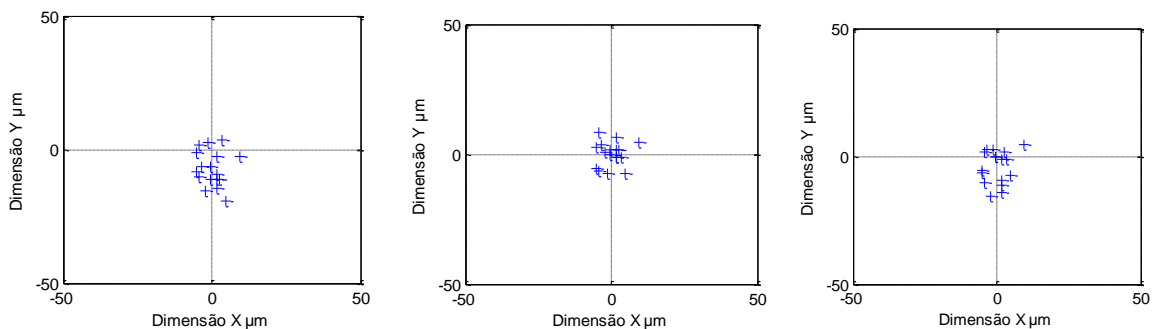


**Figura 22** – Análise de repetitividade dividida em 9 regiões.

Estes testes foram executados da seguinte maneira: a máquina foi posicionada em um ambiente com pouca variação térmica e livre de perturbações externas. Após, fixou-se o relógio comparador na mesa de trabalho do CNC e, para cada região, foi analisada a repetitividade dos eixos X e Y de maneira independente, a partir da coleta de 20 amostras para cada situação. Os resultados são apresentados nos gráficos das Figuras 23, 24 e 25. Depois de fixado o relógio comparador, enviou-se comandos para a máquina avançar e recuar por 20 vezes em cada um dos eixos.

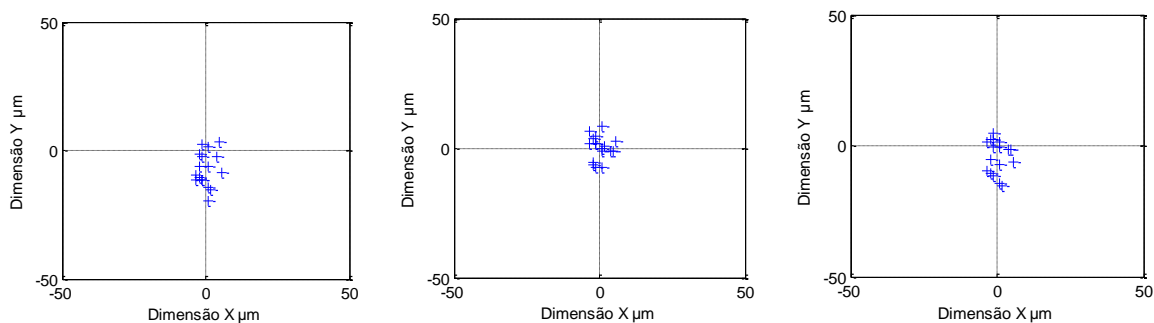


**Figura 23** – Teste de repetitividade nas regiões 1, 2 e 3. Apresentam desvios padrões de 6,4  $\mu\text{m}$ , 5,4  $\mu\text{m}$  e 6,2  $\mu\text{m}$  respectivamente.



**Figura 24** - Teste de repetitividade nas regiões 4, 5 e 6. Apresentam desvios padrões de 5,1  $\mu\text{m}$ , 4,4  $\mu\text{m}$  e 5,8  $\mu\text{m}$  respectivamente.





**Figura 25** – Teste de repetitividade nas regiões 7, 8 e 9. Apresentam desvios padrões de 6,2  $\mu\text{m}$ , 3,1  $\mu\text{m}$  e 5,2  $\mu\text{m}$  respectivamente.

A fim de avaliar o índice de repetitividade da máquina, foi realizado o cálculo de desvio padrão baseado na distribuição geométrica de cada ponto medido, de acordo com a equação X. Como resultado final, o desvio padrão obtido a partir da análise de todas as regiões foi de 8,6  $\mu\text{m}$ . Entretanto, é necessário avaliar com cuidado estes resultados, pois nesses testes a máquina não estava em operação, desconsiderando-se portanto efeitos como vibração decorrente do atrito da ferramenta com a peça, deformações dinâmicas, entre outros fatores que ocorrem com a máquina em operação.

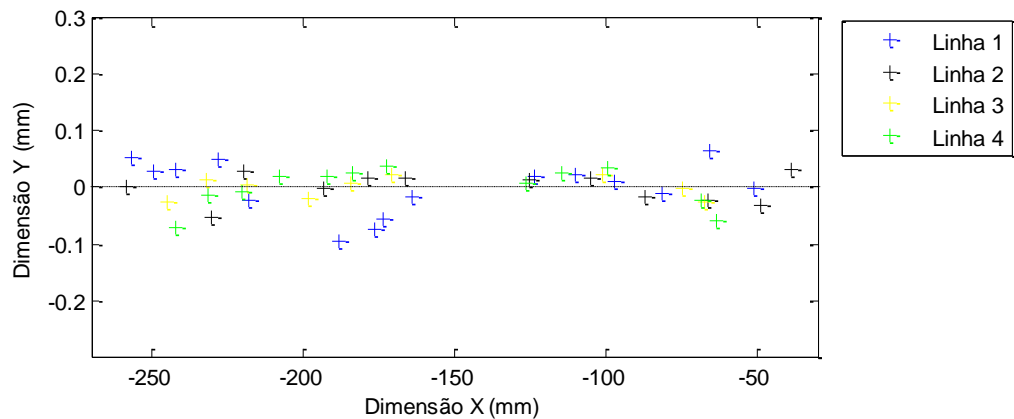
### 5.1.2 Análise de Retilidade

Para identificar a retilidade da máquina colocou-se a máquina a usinar duas peças retangulares constituídas de MDF, as peças foram usinadas para as velocidades de 300 mm/min e 500 mm/min. Depois de concluídas, as peças passaram por um processo de medição em uma máquina de medição por coordenadas, essa máquina está ilustrada na Figura 25 e possui tolerância de aproximadamente 10  $\mu\text{m}$ .



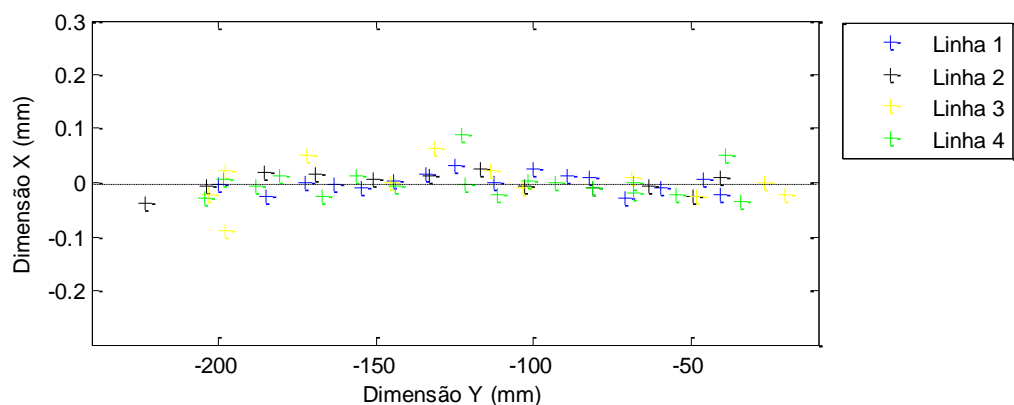
**Figura 26** – Máquina de medição tridimensional.

As medições foram realizadas nas bordas, sendo que cada peça retangular foi medidas em 60 pontos diferentes.



**Figura 27** – Teste de retilidade do eixo x apresentou desvio padrão de 0,0342 mm.

Na Figura 25 então mostrados os pontos obtidos pela máquina de medição na dimensão X, sendo que as linhas 1 e 2, são referentes às bordas do quadrado usinado na velocidade de 500 mm/min, e as linhas 3 e 4 referem-se ao segundo quadrado com velocidade de 300mm/min. sobrepondo as medições realizadas nesta dimensão pode-se notar que a máquina não apresentou nenhum erro sistemático significativo, ou seja, o erro de repetitividade neste caso foi maior que o erro de offset. Além disso percebe-se que diferentemente do teste anterior, neste caso com a máquina em operação notou-se um aumento do desvio padrão, que se devem a alguns fatores já mencionados, tais como vibração e irregularidades do material usinado. Quantificando estes resultados de desvio padrão pela distribuição dos pontos, de acordo com a Equação (2), obteve-se valor de 0,0342 mm, enquanto o experimento anterior apresentou desvio padrão de 0,0086 mm.



**Figura 28** – Teste de retilidade do eixo y apresentou desvio padrão de 0,0268mm.

Para a análise de retilidade do eixo Y, pode-se notar que assim como no caso anterior, não houve grandes variações em relações ao erro sistemático, como pode ser observado pela Figura 28. Calculando-se o desvio padrão para este caso, obteve-se o valor de 0,0268 mm, que mostrou-se inferior se comparado ao eixo x.

### 5.1.3 Análise de Perpendicularidade e Paralelismo

O procedimento para a análise de perpendicularismo foi feito utilizando-se uma placa de furos de 20x30 pontos, sendo que o espaçamento entre os pontos foi de 10 mm.

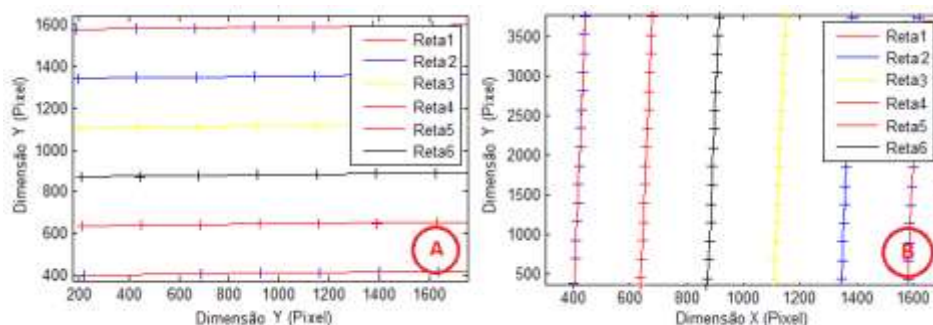
Depois de criada a sequência de comando, os dados foram enviados para a máquina, de modo a realizar um processo de furação em uma placa de MDF revestida com fórmica branca. Depois que este processo “físico” foi realizado, essa placa foi colocada em um digitalizador de duas dimensões com resolução de 1200 *dpi* (pontos por polegada). Essa resolução foi encontrada em uma máquina profissional, em uma gráfica especializada. A imagem digitalizada é apresentada na figura a seguir.



**Figura 29** – Imagem digitalizada da placa de furos utilizada para os testes.

Para validar os resultados de medições, digitalizou-se uma “forma padrão” cujas dimensões eram previamente conhecidas, para calibração e correção de eventuais distorções do processo de digitalização. Depois de digitalizada e validada a imagem, foi desenvolvido um script no Matlab para achar o centro geométrico de cada ponto e com isso obteve-se efetivamente a matriz de pontos desejada. A partir dos resultados da matriz de pontos realizou-se a análise de perpendicularidade e paralelismo.

Primeiramente, com base na matriz de pontos obtida, foi feita uma regressão linear dos dados, agrupados em linhas e colunas, de acordo com o procedimento descrito no item 3.3.3, a fim de encontrar as equações das retas horizontais (Figura 30a) e verticais (Figura 30b) que se ajustassem melhor aos dados.



**Figura 30** – Retas obtidas por regressão linear. (a) Verticais (b) Horizontais.

- **Análise de Paralelismo**

Com base nos dados da matriz calculada, foi obtido o ângulo de cada reta horizontal em relação à horizontal da imagem. Foi calculada a média desses valores, e também o desvio de cada valor em relação à média. Os resultados são apresentados na Tabela 2, onde verifica-

se que a média no eixo horizontal foi de  $0,68$  graus e o desvio padrão  $S = -0,011^\circ$ . Esse valor de  $0,68^\circ$  da média se deve essencialmente à inclinação da digitalização da própria imagem.

**Tabela 2** - Análise de inclinação das retas no sentido x.

Reta 1a	Reta 2a	Reta 3a	Reta 4a	Reta 5a	Reta 6a	Média $m_a$	Desvio Padrão (S)
$-0,68^\circ$	$-0,66^\circ$	$-0,69^\circ$	$-0,68^\circ$	$-0,69^\circ$	$-0,68^\circ$	$-0,68^\circ$	$-0,011^\circ$

Similarmente, para as retas verticais tem-se a média de inclinação com valor de  $89,30^\circ$  e desvio padrão de  $-0,007^\circ$  (Tabela 3).

**Tabela 3** - Análise de inclinação das retas no sentido y em relação à média.

Reta 1a	Reta 2a	Reta 3a	Reta 4a	Reta 5a	Reta 6a	Média $m_b$	Desvio Padrão (S)
$89,31^\circ$	$89,30^\circ$	$89,31^\circ$	$89,30^\circ$	$89,32^\circ$	$89,31^\circ$	$89,30^\circ$	$-0,007^\circ$

- **Análise de Perpendicularismo**

Para analisar perpendicularidade foi calculada a diferença entre as médias dos ângulos referentes a cada dimensão analisada (dimensão X e Y). Quantitativamente este resultado foi  $m_a - m_b = 89,99$ ; já o resultado do desvio padrão é  $S = 0,012$ , de acordo com a Equação (2), o qual foi obtido pela combinação das inclinações das retas verticais com as horizontais.

## 6 Conclusões

Para concluir este trabalho pode-se comentar a respeito dos objetivos estabelecidos, dentre eles o principal era construir uma máquina CNC com um custo inferior ao custo do mercado, o resultado disto foi positivo, tendo em vista que uma máquina do mesmo porte no Brasil custa mais de R\$ 7 mil reais, e que o protótipo construído custou pouco mais de R\$ 1.000,00 (mil reais). Porém, não se deve concluir precipitadamente que o produto é viável, já que não foram levados outros custos em consideração, como impostos, mão de obra, bem como gastos com energia elétrica, entre outros.

Já em relação à construção do protótipo, os resultados foram demorados e envolveram muito reprojeto, que fizeram com que os gastos totais ultrapassem o orçamento previsto. Além disso, para a conclusão do protótipo foi demandado um tempo de aproximadamente um ano, apesar deste dado não ser totalmente realista, pois o projeto não foi executado constantemente durante todo este tempo. Ainda assim, despendeu-se um grande tempo em relação ao previsto. Porém, se forem tomados alguns cuidados, tais como organização do processo de furação (etapa que tomou a maior parte do tempo considerando os retrabalhos) utilizando-se gabaritos de furação, bem como a terceirização de alguns serviços de torneamento e construção de gabaritos de montagem, acredita-se ser possível reduzir o tempo a menos de 5% (cinco por cento) do tempo gasto, o que tornaria o protótipo viável para manufatura.

Se por um lado o protótipo demandou muito tempo, fato já esperado devido ao projeto ser de um protótipo, por outro a elaboração do software atendeu muito bem ao outro importante objetivo estabelecido, que era o de tornar o processo de usinagem CNC uma tarefa simples para qualquer usuário. Mesmo assim, ainda existem melhorias a serem feitas nesta área.

Em relação às tolerâncias obtidas, os resultados mostraram-se satisfatórios, tendo em vista que o desvio padrão obtido para a repetitividade foi cerca de 8  $\mu\text{m}$ , o que é um resultado baixo considerando outras fresadoras de mesmo porte e custo. Já em relação à análise de retilidade, constatou-se que a máquina não apresenta erros sistemáticos, e desta forma, não existe método para corrigir os erros apresentados; ainda sobre a retilidade, pode-se aferir que os desvios padrões foram de 0,0342 mm e 0,0268 mm para os eixos X e Y, respectivamente. Como última análise, foram realizados testes de perpendicularidade e paralelismo, onde foram obtidos máximos desvios de inclinação de 0,01° para retas na direção do eixo Y, e de 0,01° para retas na direção do eixo X.

A respeito dos métodos utilizados para se obter os resultados, ou seja, a utilização do relógio comparador, da máquina de medição tridimensional e do digitalizador de duas dimensões, pode-se concluir que foram satisfatórios para cada situação na qual foram aplicados.

Em síntese, o projeto da fresadora CNC pode parecer simples, porém, apesar da utilização de componentes de alta qualidade, ainda é necessário aprimorar a precisão final da máquina. A fim de preparar o produto para comercialização, ainda é preciso otimizar a relação entre custo e qualidade.

## 6.1 Trabalhos Futuros

Como mencionado anteriormente, o protótipo apresentou tolerâncias satisfatórias, porém, em virtude do pouco tempo disponível para o projeto, não foi possível empregar um controle de qualidade mais rigoroso. A seguir citam-se algumas sugestões para aprimoramento da qualidade do projeto:

- Solicitar ao fabricante do fuso de esfera que realize os desbastes nas pontas dos mesmos em um torno CNC. Este foi um fator crucial que prejudicou os resultados de tolerância da máquina;
- Aumentar a massa da estrutura para reduzir vibrações geradas pelo processo de usinagem;

Para a parte de *software*, ainda estão faltando serem implementadas algumas partes. No *Gstudio*, faltam ser criados alguns tópicos, que estão listados a seguir:

- Ferramenta de CAD, como círculo, arco, polígonos e outras ferramentas para facilitar a criação de desenhos;
- Ferramenta de seleção de pontos e edição de propriedades;

Quanto ao *hardware*, pode-se sugerir algumas melhorias, como por exemplo:

- Realimentação da posição para corrigir problemas tais como: escorregamento de passo dos motores, identificação da posição da máquina no *startup* e também com realimentação por *encoder* é possível realizar a calibração da máquina de maneira mais automatizada.
- Adicionar um painel para controlar funções básicas da máquina, tais como um botão de parada emergencial, e um *joystick* para controlar diretamente o posicionamento da máquina sem passar pelo *software*.

## 7 Referências Bibliográficas

ABNT (2004). NBR ISO 724. **Rosca métrica ISO de uso geral - Dimensões básicas**. Rio de Janeiro: Associação Brasileira de Normas Técnicas.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NM-ISO 230-1**: Código de ensaio de máquinas-ferramenta. 1999. 83 p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NM-ISO 230-2**: Código de ensaio de máquinas-ferramenta. 2006. 32 p.

BSI. **BS EN ISO 12100:2010**: Safety of machinery. General principles for *design*. Risk assessment and risk reduction. 2010. 90 p.

BOWYER, Adrian. **REPRAP. The 3D printer that prints itself**. Disponível em: <[http://reprap.org/wiki/Main\\_Page](http://reprap.org/wiki/Main_Page)>. Acesso em: 20 nov. 2013.

CNI/COMPI. **Avaliação da conformidade**: Conhecendo e aplicando na sua empresa. 2. ed. Brasília, 2002. 78 p.

E-STORE. **Guiding Rail Shaft of Linear Movement Bearings (CCS)**. Disponível em: <<http://www.p-wholesale.com/cn-pro/19/730to1/guiding-rail-shaft-of-linear-movement-bearings-ccs-653536.html>>. Acesso em: 20 nov. 2013.

ELECTRON-SHOP. **Arduino Uno R3**. Disponível em: <<http://www.electron-shop.com/arduino-uno-r3.html>>. Acesso em: 20 nov. 2013.

INSTITUTO NACIONAL DE METROLOGIA, QUALIDADE E TECNOLOGIA; OLIVEIRA, J. C. V. de. **Vocabulário Internacional de Metrologia: conceitos fundamentais e gerais e termos associados**. Rio de Janeiro: Inmetro, 2009.

KALATEC. **Rolamento Linear Pillow Block Fechado – SMAxx-UU**. Disponível em: <[http://www.kalatec.com.br/rolamentos/SMA-Rolamento\\_linear\\_Pillow\\_Block.pdf](http://www.kalatec.com.br/rolamentos/SMA-Rolamento_linear_Pillow_Block.pdf)>. Acesso em: 20 nov. 2013.

KRAMER, T.R., Proctor, M.; Messina, E. R. **The NIST RS274NGC Interpreter - Version 3**. National Institute of Standards and Technology, Gaithersburg, MD; Ago, 2000. Disponível em: <[http://www.nist.gov/customcf/get\\_pdf.cfm?pub\\_id=823374](http://www.nist.gov/customcf/get_pdf.cfm?pub_id=823374)>. Acesso em dez 2013.

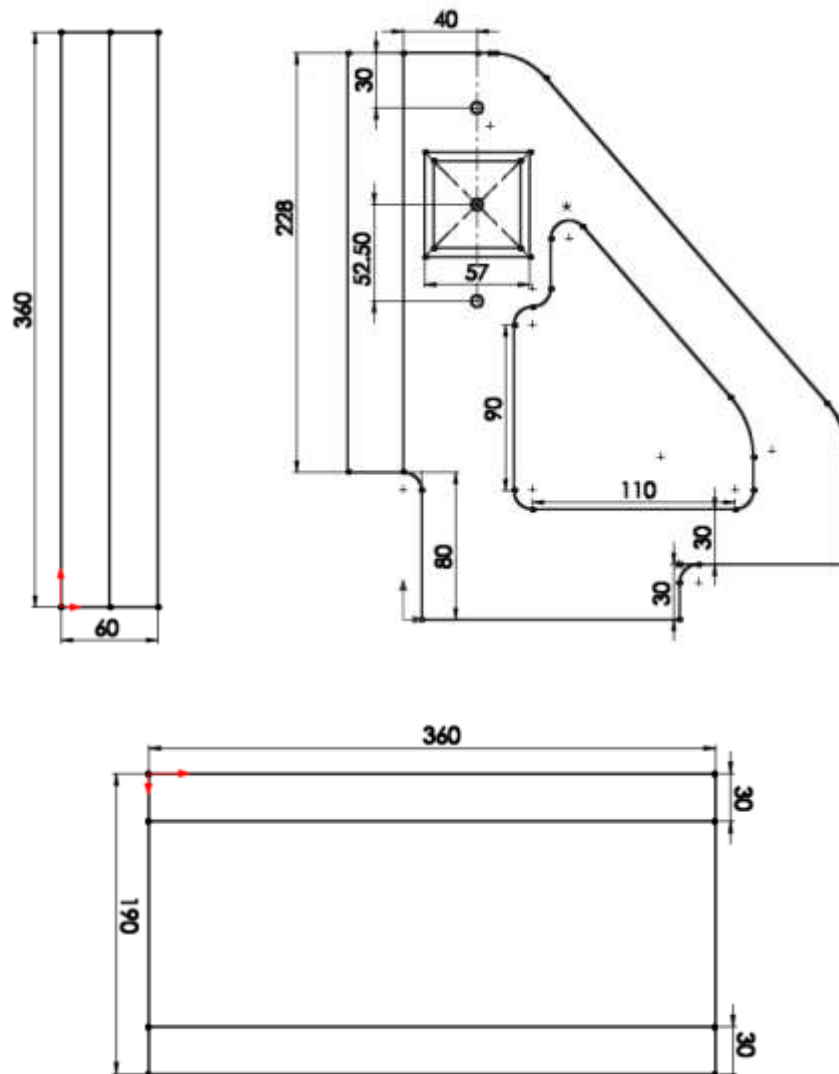
PEARCE, Joshua M. **Building Research Equipment with Free, Open-Source Hardware**. Vol.: 337 - PP. 1303-1304. DOI: 10.1126/science.1228183

SKOGRUD, Simen Svale. **GRBL: Motion control for machines that makes things**. Disponível em: <<http://bengler.no/grbl>>. Acesso em: 20 nov. 2013.

SOUSA, A. R. de. **Padrões Corporificados e a Tecnologia de Medição por Coordenadas Inovando a Qualificação Geométrica de Centros de Usinagem**. Tese - Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis, 2000. Disponível em: <<http://www.forma3d.com.br/downloads/TESE%20-%20ARS%20.pdf>>. Acesso em: 6 de dez. 2013.

STOETERAU, R. L. **Introdução ao Projeto de Máquina-Ferramentas Modernas**. 2004. Florianópolis: Universidade Federal de Santa Catarina. Disponível em: <[http://files.comunidades.net/mutcom/Projetos\\_de\\_maq\\_ferramentas.pdf](http://files.comunidades.net/mutcom/Projetos_de_maq_ferramentas.pdf)>. Acesso em: nov. 2013.

TANAKA, Marcelo Costa. **Avaliação de um Dispositivo Utilizado em Máquina de Medir por Coordenadas**. 2009. 81 f. Monografia - Engenharia de Petróleo, Departamento de Departamento de Engenharia Mecânica, Universidade Federal do Rio Grande do Norte, Natal, 2009. Disponível em: <[http://www.more.ufsc.br/tese\\_dissert/insere\\_tese\\_dissert](http://www.more.ufsc.br/tese_dissert/insere_tese_dissert)>. Acesso em: 20 nov. 2013.

**Apêndice A – Desenho da Estrutura Principal**

**Figura A.1** - Desenhos CAD no formato DXF aceito na máquina de corte de chapa por plasma (dimensões em mm).



## Apêndice B – Custos

**Tabela B1** - Custos dos materiais

Recurso	Custo	Recurso	Custo
3 x Drivers dos motores	R\$ 120,00	Chapas recortes e dobras	R\$ 100,00
3 x Acopladores	R\$ 63,00	1 x Fonte de alimentação	R\$ 100,00
1 x Fuso z	R\$ 92,00	1 x Arduino Uno	R\$ 25,00
1 x Fuso y	R\$ 115,00	1 x Arduino Mega	R\$ 36,00
1 x Fuso x	R\$ 136,00	1 x Display LCD	R\$ 30,00
4 x Rolamentos linear sfw-20mm	R\$ 132,00	1 x E-Stop Button	R\$ 5,00
6 x Rolamentos linear sfw-16mm	R\$ 156,00	1 x Joystick	R\$ 5,00
3 x Motor de Passo	R\$ 306,00	3 x Encoder	R\$ 60,00
1,1m Guia linear 16mm - Eixo x	R\$ 70,00	1,1m x Guias lineares 20mm – Eixos x	R\$ 80,00

Anexo A – Drivers dos motores

**TOSHIBA**

TB6560AHQ/AFG

TOSHIBA BiCD Integrated Circuit Silicon Monolithic

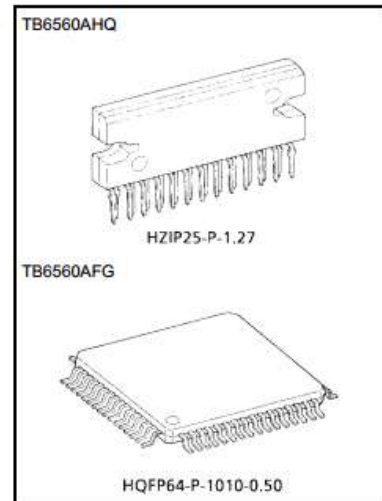
**TB6560AHQ, TB6560AFG**

PWM Chopper-Type Bipolar Driver IC for Stepping Motor Control

The TB6560AHQ/AFG is a PWM chopper-type stepping motor driver IC designed for sinusoidal-input microstep control of bipolar stepping motors. The TB6560AHQ/AFG can be used in applications that require 2-phase, 1-2-phase, 2W1-2-phase and 4W1-2-phase excitation modes. The TB6560AHQ/AFG is capable of low-vibration, high-performance forward and reverse driving of a two-phase bipolar stepping motor using only a clock signal.

**Features**

- Single-chip motor driver for sinusoidal microstep control of stepping motors
- High output withstand voltage due to the use of BiCD process: Ron (upper and lower sum) = 0.6 Ω (typ.)
- Forward and reverse rotation
- Selectable phase excitation modes (2, 1-2, 2W1-2 and 4W1-2)
- High output withstand voltage: VDSS = 40 V
- High output current: IOUT = AHQ: 3.5 A (peak)  
AFG: 2.5 A (peak)
- Packages: HZIP25-P-1.27  
HQFP64-P-1010-0.50
- Internal pull-down resistors on inputs: 100 kΩ (typ.)
- Output monitor pin: MO current (IMO (max) = 1 mA)
- Reset and enable pins
- Thermal shutdown (TSD)



Weight  
HZIP25-P-1.27: 9.86 g (typ.)  
HQFP64-P-1010-0.50: 0.26 g (typ.)

Figura A1 - Datasheet – TB6560AHQ.

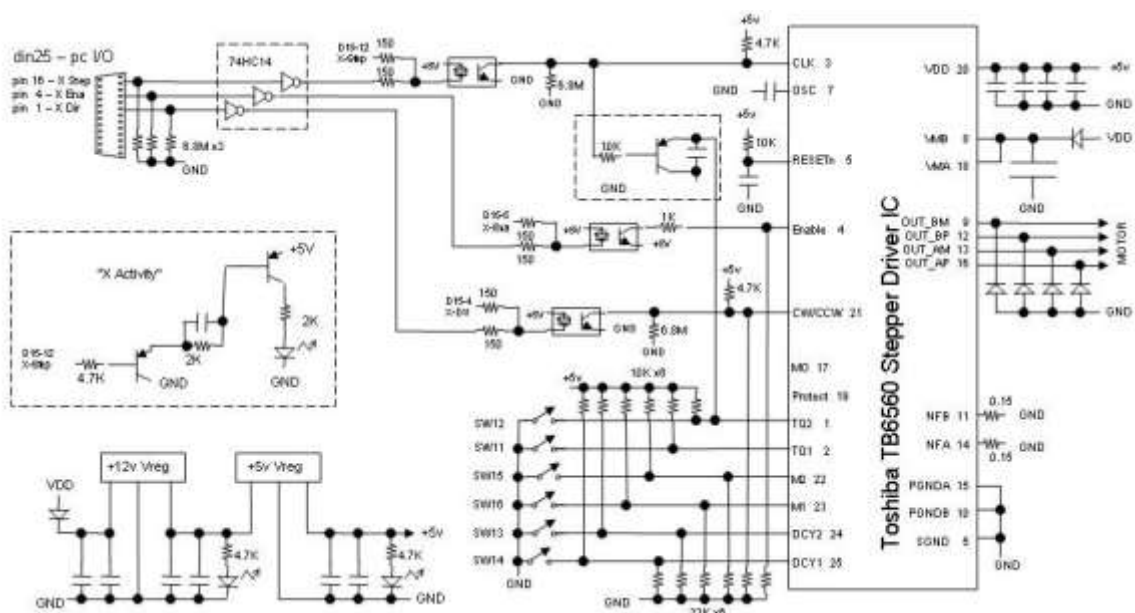


Figura A2- Esquema elétrico do circuito dos drives dos motores.