



# CNC Caseira Controlada por um Software Embarcado Open Source



Lucas Fernando da Silva, Thúlio Marcus

FUPAC - Fundação Presidente Antônio Carlos de Conselheiro Lafaiete

Engenharia de Controle e Automação

**Resumo**—Neste artigo será abordado o projeto e a construção de uma CNC (Comando Numérico Computadorizado) caseira e com baixo custo, com finalidade de trazê-la como uma ferramenta CAD (*Computer Aided Design*) e CAM (*Computer Aided Manufacturing*) furar peças em MDF (*Medium Density Fiberboard*), prototipagem de PCI (Placa de Circuito Impresso) e desenhos nos planos X, Y e Z. O projeto e construção podem ser divididos em três partes: parte mecânica, parte eletrônica e configuração dos softwares. Serão apresentados, teoricamente, tópicos relacionados a cada *hardware* e *software* envolvidos na elaboração da máquina a ser desenvolvida a partir deste artigo, como a introdução ao funcionamento do microcontrolador Arduino UNO, CNC *Shield*, *driver* A4988, motor de passo NEMA 17, módulo relé e micro retífica. Serão tratadas metodologias de produção avançadas existente no mercado como o CAD que é uma ferramenta que permite criar projetos pelo computador e o CAM que é a interação do *software* com a máquina, que a partir do desenho, gera comandos e coordenadas que a CNC compreende para o processo de trabalho nos três eixos. O projeto possibilitará ao usuário, fazer furos em MDF sem necessitar entrar em contato com a madeira e a parte ferramenta do processo preservando a integridade física do operador.

**Palavras Chave** — Arduino, Automatização, CAD, CAM, CNC, Controle, *Driver*, Fresa, *Shield*, *Software*, Usinagem.

## I. SIGLAS

3D – Três Dimensões

A – Ampere

CAD – *Computer Aided Design*

CAM – *Computer Aided Manufacturing*

CNC – Comando Numérico Computadorizado

GND – *Ground*

IHM – Interface Homem Máquina

LCD – *Liquid Crystal Display*

LED – *Light Emitting Diode*

MDF – *Medium Density Fiberboard*

NC – *Numeric Command*

PBR – Passo da Barra Rosqueada

PCI – Placa de Circuito Impresso

RPM – Rotação Por Minuto

V – Volts

## II. INTRODUÇÃO

As primeiras máquinas NC (*Numeric Command*), foram criadas na década de 50, e funcionavam através de cartões perfurados. Embora fosse um excelente sistema de fabricação, provando que reduziria custos das indústrias, os fabricantes demoraram para dar uma atenção a estas máquinas. Visando uma rápida adoção desse sistema, o Exército dos Estados

Unidos comprou 120 máquinas NC para alugar elas a algumas fábricas com intuito de familiarizar com a nova tecnologia. No final da década de 1950, o NC (*Numeric Command*) começou a se tornar mais popular mesmo com muitos fatores que dificultavam, como o fato de não existir o código G, que se trata de uma linguagem universal das máquinas CNC (Comando Numérico Computadorizado) atualmente. Naquela época, cada fabricante de máquina criava sua própria linguagem de programação. Mas foi na década de 60, onde tudo se tornou mais fácil, sendo padronizado o código G, e sendo criados alguns programas CAD (*Computer Aided Design*), onde desenhos feitos em papel eram substituídos por desenhos computacionais [1].

Nos últimos anos, a automação industrial vem crescendo de forma bastante expressiva. A produção ganha muito com este crescimento devido ao fato da automatização influenciar diretamente na qualidade dos produtos, na quantidade que se pode produzir em menor tempo e na eliminação de gastos devido à precisão de sua atuação. Tendo isso em vista, a utilização de sistemas automatizados torna-se indispensável.

A tecnologia CNC (Comando Numérico Computadorizado) é muito utilizada no meio industrial, e aplicada em diversas áreas. Um exemplo são as fresadoras CNC. Essas máquinas possuem estruturas com capacidade de produzir peças elaboradas computacionalmente através de desenhos técnicos. Máquina essa, cuja movimentação ocorre em um, dois, três ou mais eixos (lineares ou giratórios). Essas ações de movimentos são comandadas por um código G (uma linguagem de programação), padronizada pela norma ISO-1056:1975 [2].

Outros objetivos da fresadora CNC, que agregam valores em sua versatilidade, é furar, polir, fresar, retificar e uma grande quantidade de funções, permitindo gerar trabalhos com grande precisão e com uma alta velocidade. Esse trabalho se é dado devido à associação com programas de CAD (*Computer Aided Design*) que são desenhos assistidos por computador e CAM (*Computer Aided Manufacturing*) fabricação assistida por computador.

No entanto, um lado que pesa na utilização desses maquinários, é a falta de profissionais não capacitados e sem treinamentos específicos para desempenhar tais funções exigidas para operar a interface de programação IHM (Interface Homem Máquina). Outro ponto negativo é a durabilidade de algumas partes desta máquina, como a área de corte pois devido a muita produtividade, ela desgasta necessitando de trocas rápidas. Mas a maior vantagem encontra-se no custo benefício da máquina, pois ela produz em grande quantidade,

com velocidade e fazendo economia dos materiais.

### III. OBJETIVOS

O objetivo geral do trabalho é desenvolver um protótipo de uma máquina com intuito de automatizar a criação de desenhos e fazer furos em MDF com uma grande gama de finalidade como, usinar PCI e desenhar por meio de um CNC caseira, utilizando o maior número de materiais encontrado em casa ou até mesmo sucatas, a fim de torna-lo com um baixo custo e de fácil construção e utilização. Será desenvolvido um código de programação básico na linguagem C (no Arduino), para o teste dos motores a ser usado na máquina feito a partir do Arduino UNO e suas bibliotecas disponíveis, e utilizará o *software Universal Gcode Sender* como IHM (Interface Homem Máquina) para o controle da máquina, acesso a programação G (no *Gcode Sender*) e para acesso ao desenho técnico feito em outro programa computacional *SketchUp*. O desenvolvimento das peças da parte física do projeto será elaborada no *SketchUp*, onde também será feita as peças para trabalho na máquina. A CNC (Comando Numérico Computadorizado) terá uma área útil para desenhos em pequena escala por se tratar de um protótipo, porém útil para acabamentos de peças, furos em madeiras e para prototipagem em PCI.

### IV. REFERENCIAL TEÓRICO

Nesta secção será apresentada toda a fundamentação teórica para o desenvolvimento deste trabalho, assim como alguns componentes utilizados, que necessitam de um aprofundamento teórico.

#### A. Automação

A automação refere-se a um sistema automático relacionado com a aplicação de sistemas mecânicos, elétricos e eletrônicos de controle, pelo qual os mecanismos verificam seu próprio funcionamento, efetuando medições e correções sem a necessidade da interferência humana [3].

Por meio de um conjunto de técnicas, a automação pode ser aplicada sobre um processo com o objetivo de torná-lo eficiente. Através da maximização da produção com menor consumo de energia, melhores condições de segurança humana, a automação diminui a utilização da mão de obra em qualquer processo, reduz custos e aumenta a velocidade na obtenção das informações [3].

A automação é dotada de 5 componentes importantes: acionamento, sensoriamento, controle, comparadores e programas [3].

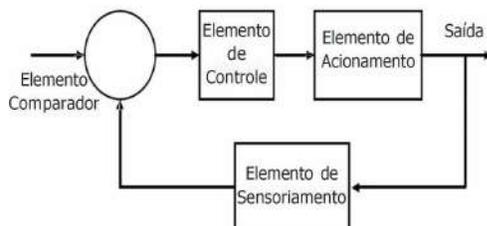


Fig. 1. Fluxograma de sistema de controle clássico . [3]

#### B. Arduino

O arduino é uma plataforma de computação física ou embarcada, ou seja, podendo interagir com seu ambiente por meio de *hardware* e *software* [4].

Graças a criação do Arduino, pode-se ter em mãos um microcontrolador desenvolvido para o uso de iniciantes e projetos acadêmicos. Com um vasto leque de recursos, ele tem opções como acionar LED's, motores de passo, módulo Ethernet, células de carga, cervo motores, displays de LCD, sensores e outros dispositivos eletrônicos [4].

O Arduino foi criado na Itália em 2005 por um professor chamado Massimo Banzi, no qual seu objetivo era lecionar eletrônica aos alunos. Antes da existência do Arduino, qualquer outro artifício que fosse usado para se aprender eletrônica pratica, seria muito caro. Então ele resolveu desenvolver uma placa para estudo. Com a ajuda de um aluno que ficou responsável pela parte da linguagem de programação do Arduino, o mesmo foi desenvolvido [5]. A fig. 2 mostra o Arduino que será utilizado no protótipo.



Fig. 2. Microcontrolador Arduino. [5]

A placa Arduino UNO é o modelo de Arduino simples, porém, proporciona recursos diversificados para utilizar em protótipos e projetos. Ela conta com 14 pinos digitais e 6 portas analógicas, e com uma tensão de entrada entre 7 a 12 volts. Na fig. 3, temos um Arduino UNO, mostrando os conectores de alimentação [6].

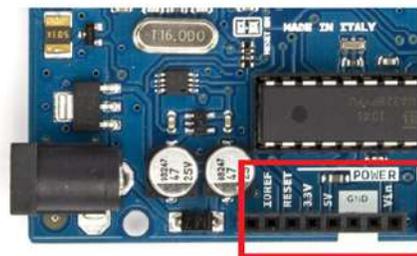


Fig. 3. Conectores para alimentação de Shields e módulos da placa Arduino UNO. [6]

**IOREF** – Fornece uma tensão de referência para que os *Shields* possam selecionar a interface apropriada, dessa forma os *Shields* que funcionam com a placa Arduino que são alimentados com 5V, ser utilizados com 3.3V possam ser utilizados em 5V [7].

**RESET** — Pino conectado com função de RESET do microcontrolador. Pode ser utilizado para um *reset* externo da placa [7].

**3.3V** — Fornece tensão de 3.3V para alimentação de alguma *Shield* e/ou módulos externos [7].

**5V** — Fornece 5V para alimentação de circuitos e *Shields* externos [7].

**GND** – Pinos de referências para aterramento (terra) [7].

**VIN** – Pino para alimentar a placa através de *Shield* ou bateria extra. Quando a placa for alimentada pelo conector *Jack* tensão da fonte estará nesse pino [7].

### C. CNC Shield V3.00

A *CNC Shield* foi projetada com o intuito de simplificar as ligações dos motores de passo, tendo nela 4 soquetes necessários para cada *driver* e ligações das bobinas dos motores de passo. Esse componente suporta alimentação de 12V a 36V, observando o componente a ser ligado nela, pois existem diversos *drivers* no mercado mas cada um necessita de diferente tensão.

O objetivo de possuir os 4 soquetes para ligação, é porque a *shield* foi desenvolvida para criação de CNC, impressora 3D ou apenas para acionamento de um ou mais motores de passo simultaneamente, representando os eixos X, Y, Z e o eixo A, que pode ser um eixo auxiliar, com objetivo de duplicar o uso de motor em algum outro eixo como X, Y ou Z.

Cada soquete é reservado da seguinte forma: um *driver* para ligação do motor de passo, lugar reservado para ligação das bobinas do motor de passo, um capacitor e pinos representando os passos do motor, onde por meio de *jumpers* são selecionado o modo de passos do motor, podendo ter diferentes formas de configurar, como passo completo, meio passo, um quarto de passo e assim por diante. Possui também a replicação do botão de *reset* do arduino, um resistor de 100Ω, pinagens auxiliares de 5V, *ground*, e outros pinos que servem para ligações de acordo com cada projeto, podendo ser ligados chaves fim de curso, relés, *coolers*, LED, ou botoeiras para acionamentos e diversos componentes eletrônicos. Isso não impede a criação desses mesmos projetos através de uma *proto-board* ou PCI, apenas reduz o número de fios, e o tamanho da montagem. A fig. 4 demonstra a *Shield* utilizada no projeto.

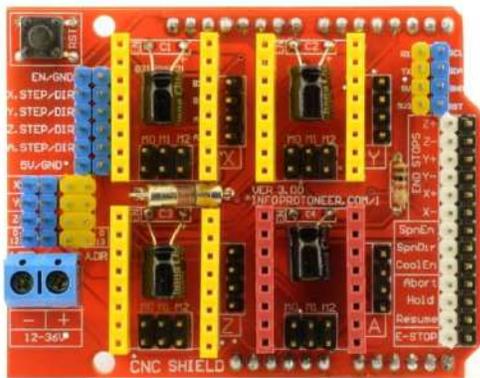


Fig. 4. CNC Shields. [8]

### D. Driver A4988

O *driver* A4988, é um controlador para motor de passo. Ele foi projetado para operar motores de passo bipolares em

diferentes modos como o passo completo (*full step*), meio passo (*Half step*), um quarto de passo, um oitavo de passo e um dezesseis avos de passo, com uma capacidade de saída de até 35V e aproximadamente 2A conforme o fabricante. Ele conta com um chip que é um ponto chave no controle do motor, pois faz o controle de corrente. Na fig. 5, temos o modelo de um *driver* A4988 e um dissipador de calor que é colocado sobre o seu chip [9].

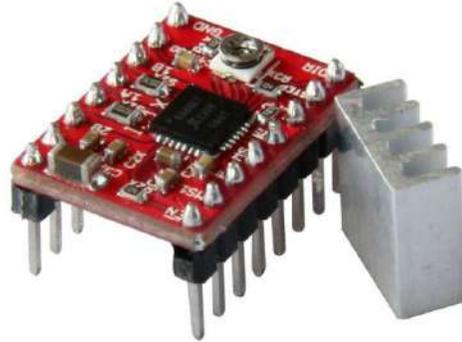


Fig. 5. Driver A4988. [9]

Em um lado do *driver*, encontra-se as portas de controle do motor e do outro lado, as conexões com o motor e os pinos de monitoramento de corrente. Todas as entradas de controle do motor, são compatíveis com os níveis de 3.3V e 5V, adequadas para o Arduino e semelhantes [10].



Fig. 6. Pinos do Driver A4988. [10]

**STEP (passo)** – um sinal com transição de baixo para alto avança um passo [10].

**DIR** – Permite alterar o sentido de rotação do motor, horário e anti-horário [10].

**ENABLE** – Um sinal baixo ativa os *drivers* do motor [10].

**RESET** – Um sinal baixo da *reset* no chip, desativando todos os *drivers* de saída, reinicia a placa (reinicia a tabela interna, desliga as saídas e ignora pulsos na entrada). Este pino deve ser mantido com tensão de Vdd para que a placa funcione. [10].

**SLEEP** – Esse pino quando negativo, desabilita alguns circuitos internos para economia de energia [10].

**Vmot e GND** – Alimentação do motor em até 35V.

**1B e 1A** – Ligar uma das bobinas do motor.

**2B e 2A** – Ligar a outra bobinas do motor.

**MS1, MS2 e MS3** – São entradas para selecionar o modo de sequencia dos passos (se vai ser passo completo, meio passo, um quarto de passo ou um oitavo de passo e um dezesseis avos de passo) [9].

**Vdd** – Alimentação do circuito de controle lógico (3V ou 5.5V).

**GND** – Terra do circuito de controle (0V).

O Ajuste de corrente de saída, é feito através de um potenciometro indicado na fig. 10, onde com um multímetro se mede a tensão de referência. Essa tensão necessita de um cálculo para encontrar a tensão essencial para o ajuste de corrente de saída. a fórmula conforme o *datasheet* do *driver* A4988 é a seguinte:

$$V_{REF} = 8 \times I_{Motor} \times R \quad (1)$$

**Vref** = Tensão de referência (Volts = V) [9].

**8** = Uma constante da formula apresentada no *datasheet* [9].

**Imotor** = Corrente do motor, conforme o *datasheet* do mesmo (*Ampere* = A) [9].

**R** = Resistor encontrado no *driver* A4988 especificado pelo fabricante no *datasheet* (*Ohm* =  $\Omega$ ) [9].

Abaixo segue a fig 7, com uma tabela demonstrando a configuração de passos onde 0 é nível baixo e 1 é nível alto dos pulsos.

Resolução Micro-passo	MS3	MS2	MS1
Passo completo	0	0	0
meio passo	0	0	1
1/4 passo	0	1	0
1/8 passo	0	1	1
1/16 passo	1	1	1

Fig. 7. Resolução de micro passo. [10]

Essa resolução tem o objetivo de determinar a angulação do passo a ser executado, visando uma melhor precisão do movimento do motor. Quanto menor a resolução, mais preciso porem, mais lento o processo.

#### E. Motor de passo Nema 17

Motor de passo é um motor elétrico que não possui comutadores e nem escovas, o que permite uma vida útil sem muito desgaste. O rotor constitui-se de um ou mais ímãs permanentes. O controle do motor é feito por intermédio de um circuito eletrônico que aciona rapidamente as bobinas que permite o girar do rotor. Os motores mais comuns são os motores que precisam de 200 passos para girar uma volta [10].

Os motores de passo utilizados no projeto foram motores bipolares da marca Nema 17. O nome bipolar é dado devido ao

fato de possuírem duas fases e normalmente um enrolamento por fase. A corrente no enrolamento precisa ser revertida para inverter um polo magnético. Por isso é necessário conhecer as especificações do motor e do *driver* a ser usado, pois um precisa atender ao outro [10]. A Fig. 7 mostra um motor de passo Nema 17.



Fig. 8. Motor de Passo Nema 17. [11]

Esse é um motor de passo robusto, sendo ele utilizado para projetos que necessitam de alta precisão e torque. Projeto como impressora 3D e CNC em geral utiliza motores Nema. Braços robóticos também utilizam esse tipo de motor [11].

De fato, o micropasso é a divisão do passo completo em várias sub-divisões do mesmo, tendo valores de corrente em cada sub-divisão. Como o passo completo está sendo dividido algumas partes para uma mesma RPM (Rotação Por Minuto), o número de passo será proporcional à resolução escolhida para trabalho do motor [12].

#### F. Módulo Relé 5

Relés são componentes eletromecânicos, ou seja, eles possuem uma parte elétrica e uma parte mecânica. Ele tem função de transformar cargas baixa em cargas maiores do que as fornecidas pelos microcontroladores [13].

Com o Módulo Relé 5V pode-se controlar lâmpadas, motores, eletrodomésticos e outros equipamentos utilizando apenas um pino de controle, já que o circuito a ser alimentado fica completamente isolado do circuito do microcontrolador. Fácil utilização com Arduino [14].

O módulo relé de um canal funciona com uma tensão de 5V, podendo acionar cargas de até 250 VAC ou 30 VDC, suportando uma corrente máxima de 10A. Possui LED indicador de energia, dois pinos de energia e um de controle, além do borne de saída com parafusos, facilitando a conexão dos equipamentos ao módulo [14].

O Módulo relé possui 3 pinos para o Arduino:

**Vcc** – 5V para alimentação do relé [13].

**IN** – Entrada de controle do relé [13].

**GND** – *Ground* [13].

E três bornes para saída:

**NO** – *Normal Open* (Normal Aberto) [13].  
**COM** – Comum [13].  
**NC** – *Normal Closed* (Normal Fechado) [13].

Na Fig. 9, temos um módulo relé 5V de um canal.

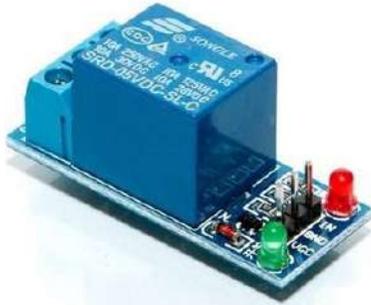


Fig. 9. Módulo Relé 5V. [13]

### G. Micro Retífica

Os produtos artesanais são conhecidos por sua beleza e simplicidade, mas nada que os tornem fáceis de serem feitos. Uma ferramenta que ajuda muito na elaboração destes trabalhos é a micro retífica, por ela ser excelente para fazer alguns cortes, furos e dar acabamento a materiais como alumínio, madeira e plástico [15].

O que torna esta ferramenta essencial é a sua versatilidade, pois pode ser usada para lixar, cortar como falado, ela é de fácil manuseio e fácil ligação e ela é indicada para trabalhos extensos [15]. Na Fig. 10, temos um modelo de uma micro retífica.



Fig. 10. Micro Retífica. [16]

No projeto, a micro retífica serve para trabalhar como a parte cortante do eixo Z da CNC, utilizando nela uma fresa para corte, furos e usinagem de PCI.

A escolha de uma fresa ao invés de usar uma broca, simplesmente se é dado pela função de cada ferramenta. A broca é utilizada para furos e a fresa para cortes laterais.

### H. Programas CAD e CAM

CAD (*Computer Aided Design*) e CAM (*Computer Aided Manufacturing*) tendo cada um desses programas parte importante no desenvolvimento de peças fabricadas com auxílio de uma máquina CNC (Comando Numérico Computadorizado).

O CAD, é uma tecnologia computadorizada focada na documentação e desenho do produto na fase de projeto, essa ferramenta permite fazer desenho gráfico bidimensional ou tridimensional de um projeto ou parte dele. Este desenho técnico fornece informações como largura, espessura e determinados detalhes como furos, ou cortes de acordo com o que se deseja. Podendo ser esse desenho para corte de produção ou apenas para ilustração de um projeto com as medidas desejadas. Abaixo na Fig. 11, é mostrado parte do projeto da CNC a ser desenvolvida citado nesse artigo, sendo criada no *Software SketchUp* a ser descrito logo abaixo na subseção 1. [17].

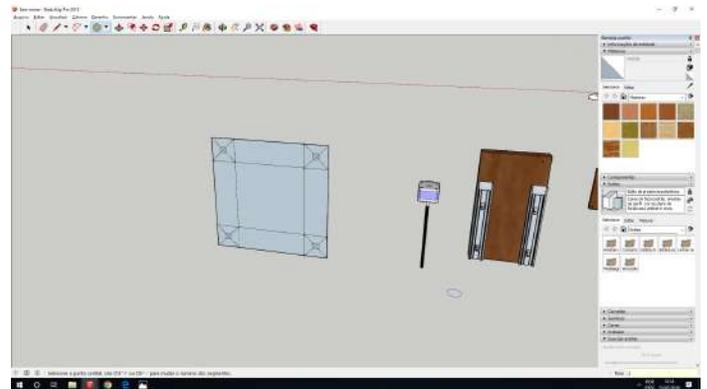


Fig. 11. Suporte do motor e eixo Z criado no *software SketchUp* [18].

O CAM, é o código por trás das máquinas que está diretamente envolvidas com a criação do produto, ou seja, máquinas controladas por computadores numéricos como fresadoras, gravadoras, soldadoras entre outros [17].

Eles são empregados após o uso do CAD, convertendo as informações fornecidas pelo desenho em coordenadas que a ferramenta deverá percorrer para fazer o trabalho [17].

#### 1. SketchUp

O *SketchUp* é um *software* CAD (*Computer Aided Manufacturing*), de fácil uso, que opera em um ambiente 3D, permitindo aos usuários criarem projetos com grande precisão. O *software* inicialmente destinado para para projetos de arquitetura e de engenharia civil tornou-se muito utilizado em outras áreas. [18].

O *SketchUp* é uma ferramenta muito intuitiva, permitindo a criação de objetos tridimensionais. E um grande motivo do seu sucesso é a objetividade do programa e por ser simples de se utilizar, diferenciando assim de outros do mercado [19].



Fig. 12. Logomarca do software SketchUp. [20]

## 2. SketchUcam

O *SketchUcam*, é uma extensão do software *SketchUp* habilitando para o usuário uma gama de ferramentas atribuindo as funções de corte, broca, fresagem CNC entre outros, gerando um *G-Code* para todas as funções, tornando possível a máquina trabalhar com as coordenadas geradas. Sendo assim a parte CAM do projeto. Na fig. 13, será apresentada as configurações iniciais do *SketchUcam* [20].

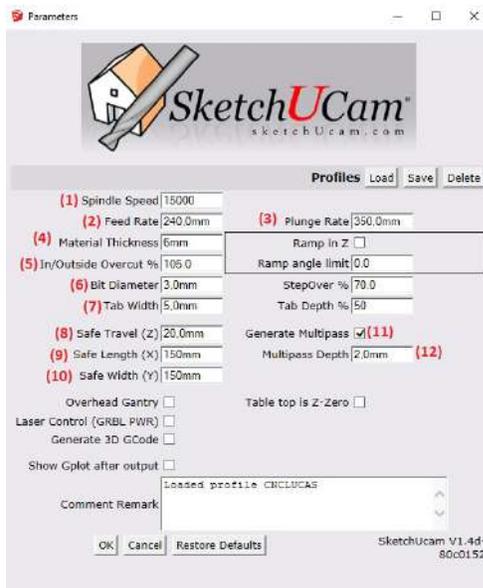


Fig. 13. Configurações do SketchUcam [21].

## 3. Universal G-Code Sender

Trata-se de um software utilizado para controle de máquinas CNC. No Arduino, é instalado o GRBL, que é um firmware (software embarcado), com alto desempenho para o controle de movimento de máquinas como uma CNC ou impressora 3D. Esta ferramenta, possibilita a leitura de *G-Code*, permitindo assim uma ampla utilização. Ele foi desenvolvido para seu código ser gravado em um micro controladores atmega328p, como o presente no Arduino UNO [22].

## 4. MakerCAM

O MakerCAM, é uma plataforma online e gratuita, que transforma imagens em código G. Ele oferece uma área para configuração dos eixos, e das ferramentas compostas na máquina, e assim, faz com que o código gerado seja

adequado a cada trabalho específico.

## 5. EAGLE 7.2.0

O EAGLE, é um software para criação de placas de circuito impresso. Ele conta com uma grande quantidade de componente eletrônico e varias bibliotecas, possibilitando assim a criação de inúmeros circuitos. Ele possui um plugin para ser gerado o código G, a partir de um circuito criado. Possibilitando assim, ser impresso em uma placa de fenolite, as trilhas de um circuito, e os furos para adaptação dos componentes eletrônicos [23].

## V. DESENVOLVIMENTO

Nesta seção, será aprofundada a parte prática do projeto, descrevendo e detalhando o passo a passo da elaboração do protótipo e a utilização dos componentes citados na fundamentação teórica. Na figura 14, é apresentado o interfacimento dos componentes principais utilizado no projeto.

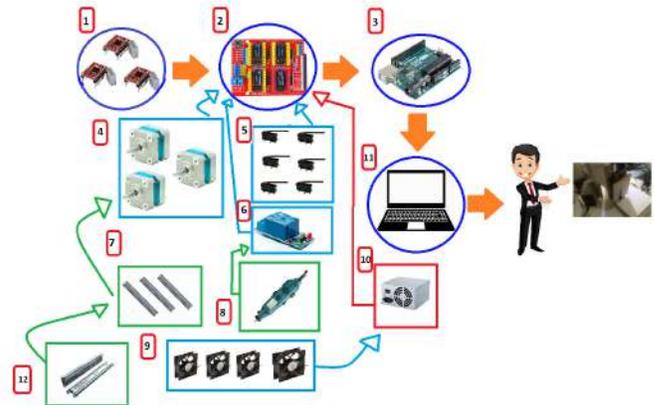


Fig. 14. Interfacimento dos componentes utilizado no projeto.

### A. Criação CAD do protótipo

Primeiramente foi desenvolvido no software *SketchUP* a parte estrutural da máquina CNC, visando calcular as medidas de cada parte do projeto e adequar-las aos componentes eletrônicos, criando assim suportes para os motores, suspensões para os eixos, a fim de alinhá-los com os eixos dos motores de passo e visando também, as medidas dos materiais que havia obtido, como as madeiras de MDF. Inspirado em particular no canal Atividade *Maker* do Rodrigo Conrado, o qual é citado na ref. [22], onde construiu uma CNC profissional dentre outros canais. Esta parte do projeto tornou-se um tanto quanto complexa, devido a falta de conhecimento no software, e a falta de experiência com similares. No entanto foi de grande valia para o trabalho, pois foi através dele, que tornou possível dimensionar o projeto. Na fig. 15, será mostrada uma imagem da criação do projeto sendo elaborados os suportes dos eixos X, Y e Z. E na fig. 16, um modelo do projeto base montado (com algumas adaptações no projeto físico como altura de acabamentos).

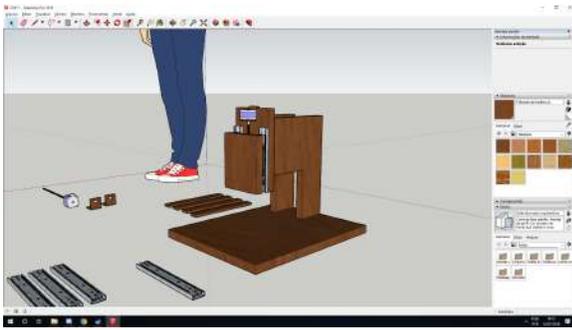


Fig. 15. Criação da CNC. em 3D. Eixo Z e suporte de motores [18].

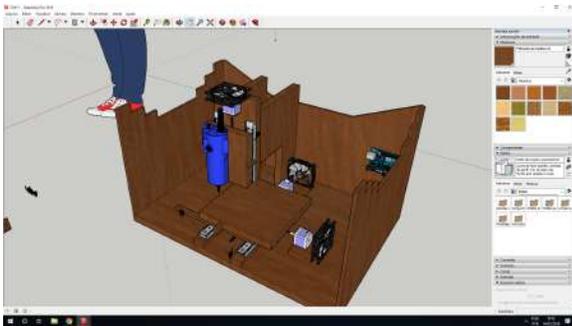


Fig. 16. Projeto base para criação do projeto físico [18].

## B. Mecânica

A partir do desenho, e das medidas elaboradas no mesmo, começaram as montagens do projeto. Começando pela parte mecânica.

Esta sessão seria a mais cara do protótipo, visto que rolamentos, guias lineares, suporte para as guias lineares, *pillow blocks*, barra rosqueada, acoplamentos flexíveis seriam partes profissionais e com preço elevado.

Tendo como uma ideia base deste artigo elaborar um protótipo de baixo custo e com materiais de fácil acesso, foram utilizados alguns componentes de custo baixo, porém com a capacidade de suprir as tarefas realizadas pelos mesmos elementos citados no paragrafo acima. Para substituir os rolamentos e as guias lineares, foram utilizados três pares de trilho telescópico *ligh* para gavetas de 25 centímetros de comprimento. Na fig. 17 está a imagem desta corredeira utilizada.



Fig. 17. Trilho de gaveta telescópico *ligh* para gavetas [23].

Os trilhos do eixo X e Y foram colocados um em cima do outro por ligação de uma chapa de MDF com 6mm, com ideia de tornar o movimento linear suave utilizando do rolamento dos trilhos para obter os movimentos desejados. Na fig. 18 demonstrado como os trilhos foram montados para torna-los eixos X e Y com a barra rosqueada de 1/4 acoplada na parte de baixo da chapa de MDF. E na fig. 19 um par de trilho foi desmontado e cortado para ser usados no eixo Z.



Fig. 18. Trilho de gaveta telescópico *ligh* para gavetas como eixo X e Y.



Fig. 19. Trilho de gaveta telescópico *ligh* para gavetas como eixo Z.

Para a movimentação dos trilhos, foram utilizadas barra rosqueada de 1/4, instaladas nos eixos conforme mostra a fig. 18. Essa barra, é colocada, uma em cada eixo, e em seguida acoplada por meio de mangueiras retrátil ao motor de passo, fazendo com que, ao girar o motor, gire também a barra rosqueada que está ligada aos eixos por meio de porcas. No entanto um cálculo deve ser apresentado pois precisa-se saber quanto um passo do motor gira em função da barra rosqueada para gerar um determinado movimento/deslocamento (passo por milímetro). Para obter esse resultado necessita do cálculo demonstrado abaixo.

$$\text{Passo/mm} = \frac{\text{config. de micropasso} \times \text{PPR}}{\text{PBR}} \quad (2)$$

**Passo/mm** – Passo por milímetro.

**Config. de Micropasso** – Configuração utilizada para rotação por passo do motor, conforme na fig. 7. Onde foi utilizado 1/4.

**PPR** – Passo por Rotação. Onde o *datasheet* descreve como 200.

**PBR** – Passo da barra rosqueada de 1/4. Seria a distância entre dois fios de rosca da barra. Utilizando um paquímetro em polegadas, e em seguida convertendo-o para milímetros, foi possível encontrar 1,27 milímetros.

Logo, a resolução para encontrar o passo por milímetro foi:

$$\begin{aligned} \text{Passo/mm} &= \frac{4 \times 200}{1.27} \\ \text{Passo/mm} &= 629,921 \end{aligned} \quad (2)$$

Depois de feito os eixos, e calculados o deslocamento por passo, foram cortadas as madeiras de MDF conforme as medidas especificadas pelo projeto CAD. Foram cortadas e montadas então, a parte física do projeto, tendo a base, o suporte para cada motor e os suportes para cada eixo específico. Na fig. 20 segue as peças de madeira, que dão origem ao corpo da CNC.



Fig. 20. Peças de MDF cortadas.

### C. Eletrônica

O sistema eletroeletrônico de uma CNC pode ser dividido em algumas partes como placa controladora, *drivers* de potência, os atuadores e ferramenta (micro retífica). Todos estes itens descritos na sessão IV deste artigo. No entanto será apresentado a seguir a junção destes componentes.

O primeiro teste foi dado por meio de uma *protoboard*, o que dificultou muito o trabalho devido a quantidade excessiva de fios, tendo em vista que a ideia era eliminar gastos desnecessários não era o plano inicial utilizar a *CNC shield*, porém para melhoria e organização do projeto ela foi adquirida. Na fig. 21 segue a primeira ligação teste de cada motor de passo ligado na *protoboard*.

Como mostrado na figura 21, a ligação de apenas um motor necessitou de grande quantidade de *jumper*s, sendo que seria necessário três vezes mais do que essa quantidade de fios pois seria utilizados mais dois motores de passo, fora as ligações de *coolers*, chaves fim de curso e relé, fez-se necessário a utilização da placa *CNC shield* para melhor organizar o projeto e por seu custo não ser alto. Na fig. 22 é apresentado a ligação.

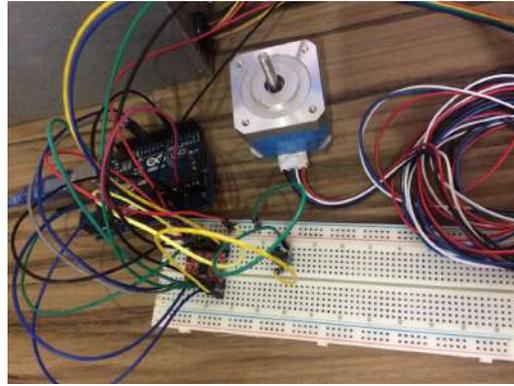


Fig. 21. Ligação do motor de passo através da *protoboard*.



Fig. 22. Ligação do motor de passo através da *CNC shield*.

Ao fazer as ligações foi exigido um cálculo, com a utilização da fórmula (1), apresentada na sub-sessão *Driver A4988*. A equação (1), com os valores de cada variável é representada abaixo.

$$\begin{aligned} V_{REF} &= 8 \times I_{Motor} \times R \\ V_{REF} &= 8 \times 1.2A \times 0.1\Omega \\ V_{REF} &= 0.96V \end{aligned} \quad (1)$$

E para o fechamento das bobinas dos motores foi seguido o que se pede no *datahseet*, conforme na fig. 23. Listado abaixo as representações de cada sigla e número para realizar estes fechamentos:

- **RED1 A** = Fio vermelho na saída 1 do motor, fechando bobina com o BLU4 Ā.
- **BLU4 Ā** = Fio azul na saída 4 do motor, fechando bobina com o RED1 A.
- **BLK3 B** = Fio preto na saída 3 do motor, fechando bobina com o WHT6 B̄.
- **WHT6 B̄** = Fio branco na saída 6 do motor, fechando bobina com BLK3 B.

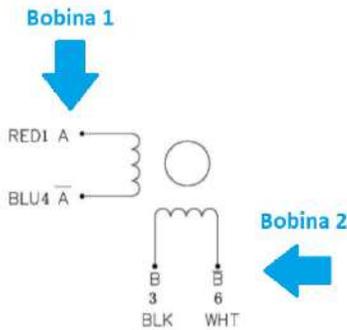


Fig. 23. Fechamento das bobinas do motor de passo [24].

Para cada motor foi utilizado um *cooler*, para resfriamento deste, como o presente na fig. 22 e um *cooler* para CNC *shield*, pois os *drivers* A4988 podem superaquecer conforme o trabalho dos motores. A ligação foi feita através de uma *protoboard* e alimentados com 12 volts, podendo ser fornecido por uma mesma fonte de computador, no caso a utilizada para alimentar a CNC *shield*, ou com alguma outra fonte paralela.

Foram utilizados também ao projeto, com uma ideia de fornecer segurança para o trabalho da máquina, cinco chaves micro *Switch* KW11-3Z-5-3T, para atuar como fim de curso e com a finalidade de desativar o funcionamento da maquina sempre que a mesma atingir os limites máximos e mínimos de sua área útil. Este componente possui três terminais sendo eles normal aberto, normal fechado e o terra.



Fig. 24. Chaves Micro Switch KW11-3Z-5-3T.

Foi feito um circuito no *software Proteus*, com a finalidade de testar um filtro anti-ruído, pois quando a máquina se movimentava, as chaves fim de curso acionavam sozinha durante o percurso. Feito o filtro, utilizando capacitores e resistores nas ligações, esses ruídos foram extintos [24].

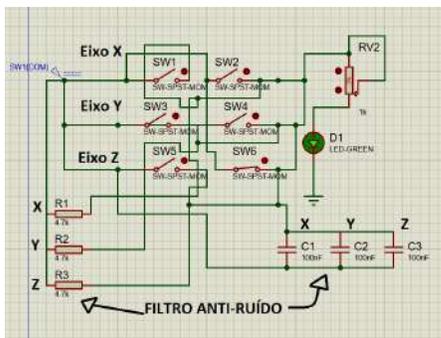


Fig. 25. Filtro anti-ruído para teste no *software Proteus* [24].

Foi também empregado um módulo relé com a finalidade ligar e desligar a micro retífica para executar algum trabalho. Conforme explicado no referencial teórico, o relé tem o objetivo de pegar uma tensão baixa e a fim de ligar a retífica que necessita de 110 volts para funcionar.

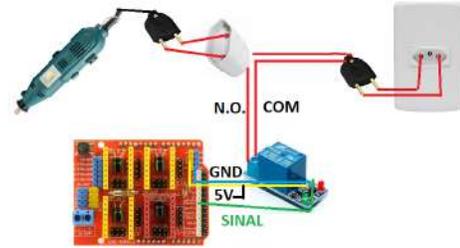


Fig. 26. Ligação do módulo relé.

E por fim, o microcontrolador Arduino UNO, como finalidade de controlar todos estes elementos eletroeletrônicos. Na fig. 24 segue a ligação em cada entrada do Arduino, e o outro lado reservado para as alimentações de 5V, 3.3V e para os *grounds*.



Fig. 27. Ligação de cada componente no Arduino.

A escolha da fonte de alimentação do projeto foi dada devido ao somatório do consumo de cada componente, tendo em vista que cada motor consome até 1.2A, utilizando três, seriam 3.6A, mais o consumo de cada *driver* e do Arduino UNO, totalizam juntos aproximadamente 1A. Sendo assim, foi utilizada uma fonte ATX com 250 Watts de potência havendo nela saídas de tensão de +3.3V, +5V, +12V. Sendo fornecida pela tensão de +12V uma corrente de aproximadamente 12A.



Fig. 28. Fonte de alimentação utilizada no protótipo .

#### D. CAD E CAM executado pelo protótipo

Esta CNC, conta a utilização de alguns *softwares* para fazer a interação homem máquina. Foram utilizados nela, o *SketchUP* como parte CAD e também o *SketchUcam* e o *Unirversal Gcode Sender* como parte CAM no projeto.

Por meio do *SketchUP*, se é criada um desenho 3D, com as medidas a serem trabalhadas pela CNC, logo em seguida no *SketchUcam* é setada as informações de trabalho, como: diâmetro da fresa, espessura do material a ser cortado, ou apenas gerando o código G do desenho a ser feito pela máquina, a ser lido em sequência pelo *Unirversal Gcode Sender*. Conforme a fig. 13, foram numerados cada item para configuração do *SketchUcam*, e abaixo será listado suas configurações:

- (1) **Spindle Speed** – Velocidade de corte da micro retífica (Padrão)
- (2) **Feed Rate** – Velocidade dos eixos X e Y, utilizando 240mm/s
- (3) **Plunge Rate** – Velocidade do eixo Z, utilizando 350mm/s
- (4) **Material Thickness** – Espessura do material a ser cortado
- (5) **In/Outside Overcut %** – Quantidade do material que será cortado em porcentagem
- (6) **Bit Diameter** – Diâmetro da fresa utilizada para os cortes
- (7) **Tab Width** – Não faz o corte do material todo, deixando uma parte da peça a ser cortada no MDF para não se desprender durante o a usinagem, 5mm
- (8) **Safe Travel Z** – Subida de segurança no eixo Z, para fresa durante o deslocamento não encostar no objeto a ser cortado, 20mm
- (9) **Safe Length** – Área de segurança no eixo X, para máquina não extrapolar os limites da parte mecânica, 150mm
- (10) **Safe Width** – Área de segurança no eixo Y, para máquina não extrapolar os limites da parte mecânica, 150mm
- (11) **Generate Multipass** – Opção para executar os cortes por camadas, e não toda a peça de uma vez
- (12) **Multipass Depth** – Quanto a fresa desce para executar corte em uma camada da peça

Em seguida ainda utilizando o *software SketchUp*, é desenhada a peça em 3D e 2D, sendo a feita em duas dimensões utilizadas para gerar o código G. Foi elaborada uma peça aleatória, para demonstrar como é gerado o código G a partir do *sketchUcan*. Na fig. 29, é demonstrada a criação do objeto.

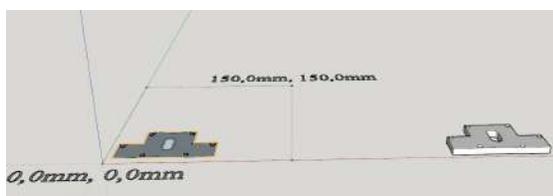


Fig. 29. Criação do suporte para motor de passo [18].

Como tratado neste artigo, a grande vantagem de se trabalhar com uma máquina CNC, é o seu potencial de produção em massa, e isso pode ser demonstrado com a criação de inúmeras peças em uma mesma demanda. Conforme a fig. 29, a mesma peça criada na fig. 30 será replicada 4 vezes para uma multi produção.

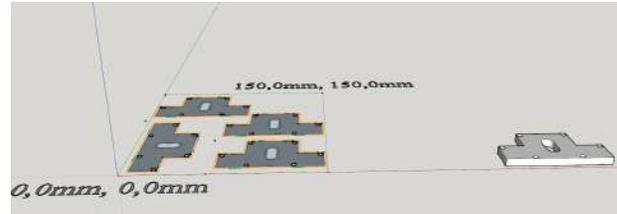


Fig. 30. Criação em massa de quatro suportes [18].

Realizadas essas etapas do desenho e das configurações do *SketchUcam*, ou de outro *software* como o *Eagle*, utilizado para fazer esta trilha na PCI, pode se gerar um arquivo em código g e ser aberto no *Universal Gcode Sender*. Na sessão apêndice, será apresentada a configuração realizada no *software* para leitura das coordenadas dos movimentos nos eixos. Utilizando o *Eagle* primeiro se cria o projeto montando o circuito como deseja, com os componentes que se precisa, e logo em seguida transfere em código G. Nas figuras 31 e 32, será apresentada o circuito e a imagem da trilha que foi gerada.

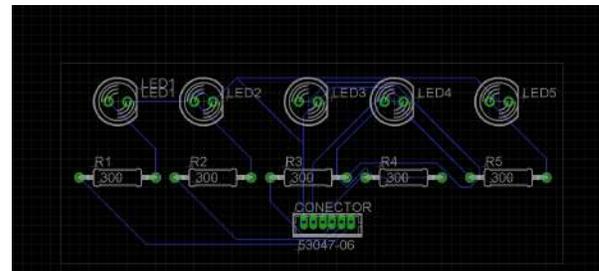


Fig. 31. Circuito de led e resistores para PCI [23].

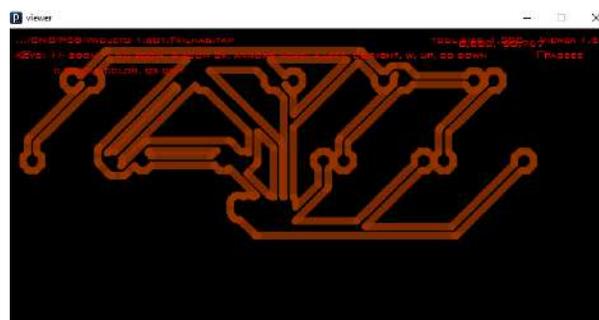


Fig. 32. Trilha gerada a partir do Circuito [23].

O *software Universal Gcode Sender* apresenta uma imagem do objeto a ser trabalhado, mostrando os caminhos que a ferramenta irá percorrer. Na Fig. 33, foi apresentado a trilha da PCI a ser usinada. A linha amarela é representação da fresa, as linhas verdes no desenho, é o caminho em que o eixo Z irá se deslocar, e as linhas brancas representam o desenho.

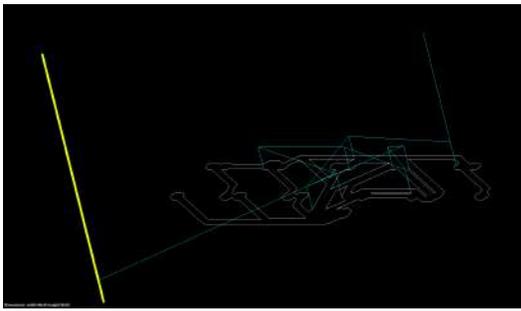


Fig. 33. Trilha para PCI [18].

A realização deste trabalho se dá diretamente ligado a uma máquina, onde há um sistema clássico de representação sobre CNC. Esse método envolve os sistemas CAD, CAM e a CNC. Na Fig. 34 será demonstrado a ordem de atuação entre eles.



Fig. 34. Sistema Clássico de funcionamento de uma CNC.

Terminado estas etapas, a CNC caseira, encontra-se em perfeito estado para o uso a qual ele foi destinado.



Fig. 35. CNC montada e pronta para utilização.

## VI. CONCLUSÃO

Este trabalho apresentou a elaboração de uma máquina CNC, implementando tecnologias para proporcionar maior eficiência, funcionalidade e conforto para os usuários e/ou empresa que adquirirem este protótipo.

A máquina desenvolvida é uma opção acessível a pesquisadores, estudantes e até mesmo para empresas. Ela permite a criação de PCI, faz furos em peças pequenas para suportes e desenhos com precisão.

O projeto atendeu às especificações desejadas, pelo fato de ter sido utilizado plataformas *open source* (código aberto), e materiais de baixo custo.

Tendo obtido resultados satisfatórios, para projetos visando algumas melhorias como aumentar sua área útil, implementar suporte para utilizar o corte a laser, para ser utilizados em gravações em madeiras, fazendo desenhos ou detalhes para melhorar os acabamentos de peças para artesanato, ou até mesmo corte de polímeros. E até mesmo, a partir desta máquina, construir uma impressora 3D.

As dificuldades encontradas neste trabalho, foram dadas devido a falta de conhecimento de algumas tecnologias e conseguir material para estudo em outros idiomas, dificuldade também para obter cada item do trabalho, pois mesmo tentando utilizar o maior número de material que já possuía, dentro da cidade onde foi projetado, não havia nenhuma loja específica para conseguir os materiais, fazendo assim uso necessário de compras *online* e até mesmo no exterior. E a maior dificuldade encontrada foi na construção e montagem da parte mecânica, devido ao não conhecimento de seus componentes.

Com a concretização deste trabalho, ele já se tornou apto a ser utilizado como um laboratório de fabricação. Para estudos e em caso de fresagem, criação de peças para CNC's futuras, ou para elaboração de peças a serem aprimoradas na própria máquina.

## REFERÊNCIAS

- [1] ALGUSTOS, João. "A história do Comando Numérico Computadorizado", 10, Jun. 2018. [Online] Disponível em: <http://www.protoprimus.com.br/maquinas-cnc-historia-comando-numerico-computadorizado/>
- [2] J.M Jaquard, "Padronização dos códigos para CNC", 16, Jul. 2018. [Online] Disponível em: <http://www.mundocnc.com.br/sabia2.php>
- [3] "O que é Automação", *Conceito e Tecnologia*, 21, Jun. 2018. [Online] Disponível em: <http://www.conceitotecnologia.com.br/automacao-oque.asp>
- [4] M. Roberts, *Arduino Básico*, vol 1. São Paulo, 2011, p.22-26.
- [5] "Programação com Arduino para Automação Industrial". 04, Mar. 2018. [Online] Disponível em: <https://pt.scribd.com/dec/199040036/Apostila-Arduino-Jhsfurno>
- [6] FILIPEFLOP. "Arduino UNO". 05, Mar. 2018. [Online] Disponível em: <https://www.filipeflop.com/produto/placa-uno-r3-cabo-usb-para-arduino/>
- [7] ] SOUZA, Fábio. "Ardino UNO". 05, Mar. 2018. [Online] Disponível em: <https://www.embarcados.com.br/arduino-uno/>
- [8] "Placa CNC shield V3.00". 10, Mar. 2018. [Online] Disponível em: <http://www.baudaeletronica.com.br/cnc-shield-v3.html>
- [9] MICROSYSTEMS, Allegro. "Driver A4988". 12, Mar 2018. [Online] Disponível em: <https://www.allegromicro.com/en/Products/Motor-Driver-And-Interface-ICs/Bipolar-Stepper-Motor-Drivers/A4988.aspx>
- [10] MURTA, Gustavo. "Motor de Passo e Driver A4988 utilizando Aduino". 11, Mar. 2018. [Online] Disponível em: <http://blog.eletragate.com/driver-a4988-motor-de-passo-usando-o-arduino/>
- [11] 4HOBBY "Motor de Passo Nema 17 4kgf.cm Impressora 3D ANET". 14, Mar. 2018. [Online] Disponível em: <https://4hobby.com.br/inicio/115-motor-de-passo-nema-17-4kgfcm-impressora-3d-anet.html>
- [12] MURTA, José Gustavo " Tutorial Arduino – Motor de Passo NEMA 17 + driver A4988". 14, Mar. 2018. [Online] Disponível em: <http://labdegaragem.com/forum/topics/tutorial-arduino-motor-de-passo-nema-17-driver-a4988?commentId=6223006%3AComment%3A524119>
- [13] LIMA, Izabelle "Módulo Relé". 16, Mar. 2018. [Online] Disponível em: <http://autocorerobotica.blog.br/modulo-rele/>
- [14] FILIPEFLOP "Módulo Relé 5V 1 Canal". 16, Mar. 2018. [Online] Disponível em: <https://www.filipeflop.com/produto/modulo-rele-5v-1-canal/>

- [15] MERLIN, Leory “Execute trabalhos detalhados e preciso com a micro retífica”. 19, Mar. 2018. [Online] Disponível em: <https://www.leroymerlin.com.br/dicas/execute-trabalhos-detalhados-e-precisos-com-a-micro-retifica>
- [16] LIVRE, Mercado. “Micro Retífica”. 20, Mar. 2018. [Online] Disponível em: [https://produto.mercadolivre.com.br/MLB-980821709-mini-micro-retifica-com-maleta-tipo-dremel-40-pecas-250w-\\_JM](https://produto.mercadolivre.com.br/MLB-980821709-mini-micro-retifica-com-maleta-tipo-dremel-40-pecas-250w-_JM)
- [17] LEÃO, Lucas. “Diferença entre CAD, CAM e CAE”. 22, Mar. 2018. [Online] Disponível em: <https://www.e3seriescenter.com/blog-engenharia-eletrica-moderna/cad-cae-e-cam-qual-a-diferenca>
- [18] RENDER “O que é SketchUp”. 22, Mar. 2018. [Online] Disponível em: <http://blog.render.com.br/diversos/o-que-e-o-sketchup/>
- [19] CAD, Plataforma. “O que é SketchUp”. 22, Mar. 2018. [Online] Disponível em: <https://www.plataformacad.com/o-que-e-sketchup/>
- [20] RENDER “SketchUp Logomarca”. 22, Mar. 2018. [Online] Disponível em: [https://deko-rady.pl/porady-projektanta-wnetrz/google-sketchup-program-do-projektowania-wnetrz\\_3754/attachment/sketchup-logo](https://deko-rady.pl/porady-projektanta-wnetrz/google-sketchup-program-do-projektowania-wnetrz_3754/attachment/sketchup-logo)
- [21] BOYZ, Phlat. “PhlatBoyz ScketchUcam(PhlatScripT)”. 25, Mar. 2018. [Online] Disponível em: <http://swarfer.github.io/sketchucam/index.html>
- [22] CONRADO, Rodrigo. “Grbl, o que é? Pra que serve?”. 04, Abr. 2018. [Online] Disponível em: <http://www.atividademaker.com.br/upload/grbl/AtividadeMakerGrbl09j.pdf>
- [23] AUTODESK. “*software Eagle*”. 26, Nov. 2018. [Online] Disponível em: <https://www.autodesk.com/products/eagle/overview>
- [24] MAGAZILLA. “*software Proteus*”. 26, Nov. 2018. [Online] Disponível em: <http://webeletronica.blogspot.com/p/proteus.html>
- [25] LIVRE, Mercado. “Trilho de gaveta telescópico *ligh* para gavetas?”. 04, Abr. 2018. [Online] Disponível em: [https://produto.mercadolivre.com.br/MLB-828634214-trilho-telescopico-corredica-light-para-gaveta-25cm-par-30kg-\\_JM](https://produto.mercadolivre.com.br/MLB-828634214-trilho-telescopico-corredica-light-para-gaveta-25cm-par-30kg-_JM)
- [26] ELETRÔNICA, Baú. “Motor de Passo NEMA 17 - 2,5 kgf.cm / 1,2A - Action?”. 04, Abr. 2018. [Online] Disponível em: [https://produto.mercadolivre.com.br/MLB-828634214-trilho-telescopico-corredica-light-para-gaveta-25cm-par-30kg-\\_JM](https://produto.mercadolivre.com.br/MLB-828634214-trilho-telescopico-corredica-light-para-gaveta-25cm-par-30kg-_JM)

APÊNDICE A  
CÓDIGO FONTE DESENVOLVIDO NA IDE DO ARDUINO PARA O TESTE DOS MOTORES

```

1
2 // Codigo para ligacao inicial motor d passo.
3
4 #include <AccelStepper.h>
5
6 int ac_motor = 100;
7
8 int vel_mot = 100;
9 int horario = 0;
10
11 int antihorario = 0;
12 int num = 0;
13
14 // Entrada ENABLE
15
16
17 int pino_enable = 10;
18
19
20 // Pinos STEP (passo) e DIR (direção)
21
22 AccStepper motor1(1,7,4 );
23
24
25 void setup( )
26
27 {
28
29 Serial.begin(9600);
30 pinMode(pino_enable, OUTPUT);
31
32 // Inicializando Motor de Passo
33
34 motor1.setMaxSpeed(vel_mot);
35 motor1.setSpeed(vel_mot);
36 motor1.setAcceleration(ac_motor);
37
38 Serial.println("Digite 1 para sentido horario, 2 sentido anti-horario ou 3 para parar o motor
39 e clique em ENVIAR");
40
41 }
42
43 void loop()
44 {
45
46 // Esperando o usuário digitar
47
48 if (Serial.available() > 0)
49 {
50
51 num = Serial.read();
52
53 {
54
55 if (num == '1')
56 {
57
58 Serial.println("Numero 1 - Rotacionando o motor para o sentido horario.");
59 digitalWrite(pino_enable, LOW);
60 horario = 1;
61 antihorario = 0;
62 }
63
64 if (num == '2')
65 {
66
67 Serial.println("Numero 2 - Rotacionando o motor para o sentido anti-horario.");
68 digitalWrite(pino_enable, LOW);
69 horario = 0;
70 antihorario = 1;
71 }
72
73 }
74
75 }
76
77 }
78
79 }
80
81 }
82
83 }
84
85 }
86
87 }
88
89 }
90
91 }
92
93 }
94
95 }
96
97 }
98
99 }
100
101 }
102
103 }
104
105 }
106
107 }
108
109 }
110
111 }
112
113 }
114
115 }
116
117 }
118
119 }
120
121 }
122
123 }
124
125 }
126
127 }
128
129 }
130
131 }
132
133 }
134
135 }
136
137 }
138
139 }
140
141 }
142
143 }
144
145 }
146
147 }
148
149 }
150
151 }
152
153 }
154
155 }
156
157 }
158
159 }
160
161 }
162
163 }
164
165 }
166
167 }
168
169 }
170
171 }
172
173 }
174
175 }
176
177 }
178
179 }
180
181 }
182
183 }
184
185 }
186
187 }
188
189 }
190
191 }
192
193 }
194
195 }
196
197 }
198
199 }
200
201 }
202
203 }
204
205 }
206
207 }
208
209 }
210
211 }
212
213 }
214
215 }
216
217 }
218
219 }
220
221 }
222
223 }
224
225 }
226
227 }
228
229 }
230
231 }
232
233 }
234
235 }
236
237 }
238
239 }
240
241 }
242
243 }
244
245 }
246
247 }
248
249 }
250
251 }
252
253 }
254
255 }
256
257 }
258
259 }
260
261 }
262
263 }
264
265 }
266
267 }
268
269 }
270
271 }
272
273 }
274
275 }
276
277 }
278
279 }
280
281 }
282
283 }
284
285 }
286
287 }
288
289 }
290
291 }
292
293 }
294
295 }
296
297 }
298
299 }
300
301 }
302
303 }
304
305 }
306
307 }
308
309 }
310
311 }
312
313 }
314
315 }
316
317 }
318
319 }
320
321 }
322
323 }
324
325 }
326
327 }
328
329 }
330
331 }
332
333 }
334
335 }
336
337 }
338
339 }
340
341 }
342
343 }
344
345 }
346
347 }
348
349 }
350
351 }
352
353 }
354
355 }
356
357 }
358
359 }
360
361 }
362
363 }
364
365 }
366
367 }
368
369 }
370
371 }
372
373 }
374
375 }
376
377 }
378
379 }
380
381 }
382
383 }
384
385 }
386
387 }
388
389 }
390
391 }
392
393 }
394
395 }
396
397 }
398
399 }
400
401 }
402
403 }
404
405 }
406
407 }
408
409 }
410
411 }
412
413 }
414
415 }
416
417 }
418
419 }
420
421 }
422
423 }
424
425 }
426
427 }
428
429 }
429
430 }
431
432 }
433
434 }
435
436 }
437
438 }
439
440 }
440
441 }
441
442 }
442
443 }
443
444 }
444
445 }
445
446 }
446
447 }
447
448 }
448
449 }
449
450 }
450
451 }
451
452 }
452
453 }
453
454 }
454
455 }
455
456 }
456
457 }
457
458 }
458
459 }
459
460 }
460
461 }
461
462 }
462
463 }
463
464 }
464
465 }
465
466 }
466
467 }
467
468 }
468
469 }
469
470 }
470
471 }
471
472 }
472
473 }
473
474 }
474
475 }
475
476 }
476
477 }
477
478 }
478
479 }
479
480 }
480
481 }
481
482 }
482
483 }
483
484 }
484
485 }
485
486 }
486
487 }
487
488 }
488
489 }
489
490 }
490
491 }
491
492 }
492
493 }
493
494 }
494
495 }
495
496 }
496
497 }
497
498 }
498
499 }
499
500 }
500
501 }
501
502 }
502
503 }
503
504 }
504
505 }
505
506 }
506
507 }
507
508 }
508
509 }
509
510 }
510
511 }
511
512 }
512
513 }
513
514 }
514
515 }
515
516 }
516
517 }
517
518 }
518
519 }
519
520 }
520
521 }
521
522 }
522
523 }
523
524 }
524
525 }
525
526 }
526
527 }
527
528 }
528
529 }
529
530 }
530
531 }
531
532 }
532
533 }
533
534 }
534
535 }
535
536 }
536
537 }
537
538 }
538
539 }
539
540 }
540
541 }
541
542 }
542
543 }
543
544 }
544
545 }
545
546 }
546
547 }
547
548 }
548
549 }
549
550 }
550
551 }
551
552 }
552
553 }
553
554 }
554
555 }
555
556 }
556
557 }
557
558 }
558
559 }
559
560 }
560
561 }
561
562 }
562
563 }
563
564 }
564
565 }
565
566 }
566
567 }
567
568 }
568
569 }
569
570 }
570
571 }
571
572 }
572
573 }
573
574 }
574
575 }
575
576 }
576
577 }
577
578 }
578
579 }
579
580 }
580
581 }
581
582 }
582
583 }
583
584 }
584
585 }
585
586 }
586
587 }
587
588 }
588
589 }
589
590 }
590
591 }
591
592 }
592
593 }
593
594 }
594
595 }
595
596 }
596
597 }
597
598 }
598
599 }
599
600 }
600
601 }
601
602 }
602
603 }
603
604 }
604
605 }
605
606 }
606
607 }
607
608 }
608
609 }
609
610 }
610
611 }
611
612 }
612
613 }
613
614 }
614
615 }
615
616 }
616
617 }
617
618 }
618
619 }
619
620 }
620
621 }
621
622 }
622
623 }
623
624 }
624
625 }
625
626 }
626
627 }
627
628 }
628
629 }
629
630 }
630
631 }
631
632 }
632
633 }
633
634 }
634
635 }
635
636 }
636
637 }
637
638 }
638
639 }
639
640 }
640
641 }
641
642 }
642
643 }
643
644 }
644
645 }
645
646 }
646
647 }
647
648 }
648
649 }
649
650 }
650
651 }
651
652 }
652
653 }
653
654 }
654
655 }
655
656 }
656
657 }
657
658 }
658
659 }
659
660 }
660
661 }
661
662 }
662
663 }
663
664 }
664
665 }
665
666 }
666
667 }
667
668 }
668
669 }
669
670 }
670
671 }
671
672 }
672
673 }
673
674 }
674
675 }
675
676 }
676
677 }
677
678 }
678
679 }
679
680 }
680
681 }
681
682 }
682
683 }
683
684 }
684
685 }
685
686 }
686
687 }
687
688 }
688
689 }
689
690 }
690
691 }
691
692 }
692
693 }
693
694 }
694
695 }
695
696 }
696
697 }
697
698 }
698
699 }
699
700 }
700
701 }
701
702 }
702
703 }
703
704 }
704
705 }
705
706 }
706
707 }
707
708 }
708
709 }
709
710 }
710
711 }
711
712 }
712
713 }
713
714 }
714
715 }
715
716 }
716
717 }
717
718 }
718
719 }
719
720 }
720
721 }
721
722 }
722
723 }
723
724 }
724
725 }
725
726 }
726
727 }
727
728 }
728
729 }
729
730 }
730
731 }
731
732 }
732
733 }
733
734 }
734
735 }
735
736 }
736
737 }
737
738 }
738
739 }
739
740 }
740
741 }
741
742 }
742
743 }
743
744 }
744
745 }
745
746 }
746
747 }
747
748 }
748
749 }
749
750 }
750
751 }
751
752 }
752
753 }
753
754 }
754
755 }
755
756 }
756
757 }
757
758 }
758
759 }
759
760 }
760
761 }
761
762 }
762
763 }
763
764 }
764
765 }
765
766 }
766
767 }
767
768 }
768
769 }
769
770 }
770
771 }
771
772 }
772
773 }
773
774 }
774
775 }
775
776 }
776
777 }
777
778 }
778
779 }
779
780 }
780
781 }
781
782 }
782
783 }
783
784 }
784
785 }
785
786 }
786
787 }
787
788 }
788
789 }
789
790 }
790
791 }
791
792 }
792
793 }
793
794 }
794
795 }
795
796 }
796
797 }
797
798 }
798
799 }
799
800 }
800
801 }
801
802 }
802
803 }
803
804 }
804
805 }
805
806 }
806
807 }
807
808 }
808
809 }
809
810 }
810
811 }
811
812 }
812
813 }
813
814 }
814
815 }
815
816 }
816
817 }
817
818 }
818
819 }
819
820 }
820
821 }
821
822 }
822
823 }
823
824 }
824
825 }
825
826 }
826
827 }
827
828 }
828
829 }
829
830 }
830
831 }
831
832 }
832
833 }
833
834 }
834
835 }
835
836 }
836
837 }
837
838 }
838
839 }
839
840 }
840
841 }
841
842 }
842
843 }
843
844 }
844
845 }
845
846 }
846
847 }
847
848 }
848
849 }
849
850 }
850
851 }
851
852 }
852
853 }
853
854 }
854
855 }
855
856 }
856
857 }
857
858 }
858
859 }
859
860 }
860
861 }
861
862 }
862
863 }
863
864 }
864
865 }
865
866 }
866
867 }
867
868 }
868
869 }
869
870 }
870
871 }
871
872 }
872
873 }
873
874 }
874
875 }
875
876 }
876
877 }
877
878 }
878
879 }
879
880 }
880
881 }
881
882 }
882
883 }
883
884 }
884
885 }
885
886 }
886
887 }
887
888 }
888
889 }
889
890 }
890
891 }
891
892 }
892
893 }
893
894 }
894
895 }
895
896 }
896
897 }
897
898 }
898
899 }
899
900 }
900
901 }
901
902 }
902
903 }
903
904 }
904
905 }
905
906 }
906
907 }
907
908 }
908
909 }
909
910 }
910
911 }
911
912 }
912
913 }
913
914 }
914
915 }
915
916 }
916
917 }
917
918 }
918
919 }
919
920 }
920
921 }
921
922 }
922
923 }
923
924 }
924
925 }
925
926 }
926
927 }
927
928 }
928
929 }
929
930 }
930
931 }
931
932 }
932
933 }
933
934 }
934
935 }
935
936 }
936
937 }
937
938 }
938
939 }
939
940 }
940
941 }
941
942 }
942

```

```
66 if (num == '3')
67 {
68 Serial.println("Numero 3 - Parando motor.");
69 horario = 0;
70 antihorario = 0;
71 motor1.moveTo(0);
72
73 digitalWrite(pino_enable, HIGH);
74 }
75
76 }
77
78 }
79
80
81 // Rotaciona o motor para o sentido horario
82
83 if (horario == 1)
84 {
85 motor1.moveTo(10000);
86 }
87
88 // Rotaciona o motor para o sentido anti-horario
89
90 if (antihorario == 1)
91 {
92 motor1.moveTo(-10000);
93 }
94 // Comando para acionar o motor
95
96 motor1.run();
97 }
```

APÊNDICE B  
CONFIGURAÇÃO DESENVOLVIDO NO *Software Universal G-Code Sender*

\$0=10 (step pulse, usec) //Tamanho do pulso em micro segundos  
 \$1=25 (step idle delay, msec) //Atraso na Inatividade do Motor  
 \$2=0 (step port invert mask:00000000) //Inversão do sinal dos Pulsos  
 \$3=0 (dir port invert mask:00000000) //Inversão da Direção do Motor  
 \$4=0 (step enable invert, bool) //Inverte o *Enable* do Controlador (desativa com nível alto e ativa com nível baixo)  
 \$5=1 (limit pins invert, bool) //Inverte os Limites  
 \$6=0 (probe pin invert, bool) //Inverte o Probe  
 \$10=3 (status report mask:00000011) //Relatório de Status (informação com as coordenadas da máquina)  
 \$11=0.010 (junction deviation, mm) //Desvio de Junção (controle de aceleração)  
 \$12=0.002 (arc tolerance, mm) //Tolerância de Arco (padrão)  
 \$13=0 (report inches, bool) //Relatórios em Polegadas (Usar 0 para mm, 1 para polegadas)  
 \$20=0 (soft limits, bool) //Limites *software* (previne que a máquina ultrapasse os limites de funcionamento)  
 \$21=1 (hard limits, bool) //Limites do Hardware (utiliza da chave fim de curso)  
 \$22=1 (homing cycle, bool) //Ciclo de Hoaming (Usado para localizar a posição de trabalho)  
 \$23=3 (homing dir invert mask:00000000) //Inverter a Direção de *Hoaming*  
 \$24=60.000 (homing feed, mm/min) //*Hoaming Feed* (velocidade até encostar na chave fim de curso)  
 \$25=500.000 (homing seek, mm/min) //*Hoaming Seel* (velocidade para procurar a chave fim de curso)  
 \$26=250 (homing debounce, msec) //*Hoaming Debouce* (sinal alto e baixo, para conferir se o sinal foi realmente do fim de curso e não um ruído)  
 \$27=6.000 (homing pull-off, mm) //*Hoaming Pull-Off* (após acionar o fim de curso, ele volta um pouco para evitar de enconstar novamente acidentalmente)  
 \$100=629.921 (x, step/mm) //Passos por milímetros eixo X (calculados pela equação (2) deste artigo)  
 \$101=629.921 (y, step/mm) //Passos por milímetros eixo Y (calculados pela equação (2) deste artigo)  
 \$102=629.921 (z, step/mm) //Passos por milímetros eixo Z (calculados pela equação (2) deste artigo)  
 \$110=420.000 (x max rate, mm/min) //Taxa máxima que o eixo X pode se mover  
 \$111=420.000 (y max rate, mm/min) //Taxa máxima que o eixo Y pode se mover  
 \$112=420.000 (z max rate, mm/min) //Taxa máxima que o eixo Z pode se mover  
 \$120=10.000 (x accel, mm/sec<sup>2</sup>) //Aceleração eixo X  
 \$121=10.000 (y accel, mm/sec<sup>2</sup>) //Aceleração eixo Y  
 \$122=10.000 (z accel, mm/sec<sup>2</sup>) //Aceleração eixo Z  
 \$130=150.000 (x max travel, mm) //Tamanho da Área Útil eixo X  
 \$131=150.000 (y max travel, mm) //Tamanho da Área Útil eixo Y  
 \$132=50.000 (z max travel, mm) //Tamanho da Área Útil eixo Z

## APÊNDICE C

### CONFIGURAÇÕES ARDUINO UNO COM GRBL

```

1
2 #ifndef config_h
3
4 #define config_h
5 #include "grbl.h" // Compatibilidade com IDE do Arduino.
6
7 #define DEFAULTS_GENERIC // Utilizado para redefinir o EEPROM, quando necessario.
8
9 // Taxa de transferencia de informacoes.
10 #define BAUD_RATE 115200
11
12 // Mapeamento do microcontrolador e seu processador utilizado.
13 #define CPU_MAP_ATMEGA328P
14
15 // Mensagens de alerta e ajuda em caso de erros acontecido devido a falhas mecanicas ou
16 // tecnicas.
17 #define CMD_STATUS_REPORT '?'
18 #define CMD_FEED_HOLD '!'
19 #define CMD_CYCLE_START '~'
20 #define CMD_RESET 0x18 // ctrl-x.
21 #define CMD_SAFETY_DOOR '@'
22
23 // Em caso de ativacao do homing automatico, o GRBL começa com a maquina bloqueada
24 #define HOMING_INIT_LOCK
25
26 // Ao executar o Homing, faz uma busca de informacoes onde ele se encontra, e retorna
27 // executando nas ordens abaixo:
28
29 #define HOMING_CYCLE_0 (1<<Z_AXIS) // Primeiro move o eixo Z para evitar dano aos materiais em
30 // caso de ferramenta
31
32 #define HOMING_CYCLE_1 ((1<<X_AXIS)|(1<<Y_AXIS)) // Em seguida move os eixos X e Y
33 // simultaneamente
34
35 // O sistema de acionamento dos limites devem ser acionados mais de uma vez, para prevenir erro
36 #define N_HOMING_LOCATE_CYCLE 1
37
38 // Bloco de comando para inicializacao do GRBL, conforme a linha 6.
39 #define N_STARTUP_LINE 2
40
41 //
42
43 #define N_DECIMAL_COORDVALUE_INCH 4 // Valor da coordenada ou posicao em polegadas (Nao
44 // utilizado)
45 #define N_DECIMAL_COORDVALUE_MM 3 // Valor da coordenada ou posicao em milimetro (Utilizado)
46 #define N_DECIMAL_RATEVALUE_INCH 1 // Taxa de velocidade polegadas/minuto (Nao Utilizado)
47 #define N_DECIMAL_RATEVALUE_MM 0 // Taxa de velocidade milimetros/minuto (Utilizado)
48 #define N_DECIMAL_SETTINGVALUE 3 // Valores decimais (Padrao)
49
50 //
51
52 // Apos concluir um trabalho, imprime-se uma mensagem na tela
53 #define MESSAGE_PROBE_COORDINATES
54
55 // Utilizado para atrado do spindle (retifica utilizada)
56 #define SAFETY_DOOR_SPINDLE_DELAY 4000
57 #define SAFETY_DOOR_COOLANT_DELAY 1000
58
59 // O modulo rele utilizado funciona invertido, quando sinal alto desliga o spindle, e quando
60 // baixo ativa o spindle.
61 #define INVERT_SPINDLE_ENABLE_PIN
62
63 // Tempo de aceleracao para movimentacao da maquina utilizada
64 #define ACCELERATION_TICKS_PER_SECOND 100
65
66 // Sincronizacao dos eixos para eliminar ruidos e movimentos nao desejados

```

```

61 #define ADAPTIVE_MULTI_AXIS_STEP_SMOOTHING
62
63 // Escolher em qual eixo encontra-se a ferramenta para seguranca da maquina
64 #define TOOL_LENGTH_OFFSET_AXIS Z_AXIS
65
66 // Usado para definir as rotacoes maxima e minma do spindle (No caso ligado e desligado)
67 #define SPINDLE_MAX_RPM 1000.0 // Rotacao maxima do spindle (Para 100%).
68 #define SPINDLE_MIN_RPM 0.0 // Rotacao minima do spindle (Para 0%)
69
70
71 // Definir a velocidade de arranque da maquina (Deixar 0.0 para nao ter arranque antes de eixo
// ferramenta se mover para posicao de seguranca)
72 #define MINIMUM_JUNCTION_SPEED 0.0
73
74
75 // Definir a taxa de alimentacao a ser usada nos movimentos
76 #define MINIMUM_FEED_RATE 1.0
77
78 // Ajuste ao calculo de passos por milimetro, colocado na configuracao $100,101 e 102 do
// Apendice B
79 #define N_ARC_CORRECTION 12
80
81
82 // Direcao circular dos motores, para trabalharem em sincronia
83 #define ARC_ANGULAR_TRAVEL_EPSILON 5E-7
84
85 // Tempo de atraso para acoes
86 #define DWELL_TIME_STEP 50 // milissegundos
87
88 // Confirmar configuracao da maquina com a configuracao GRBL
89 #ifndef HOMING_CYCLE_0
90 #error "Requer HOMING_CYCLE_0 nao definido."
91 #endif
92
93 #if defined(USE_SPINDLE_DIR_AS_ENABLE_PIN) && !defined(VARIABLE_SPINDLE)
94 #error "USE_SPINDLE_DIR_AS_ENABLE_PIN so pode ser usado com VARIABLE_SPINDLE ativado"
95 #endif
96
97 #if defined(USE_SPINDLE_DIR_AS_ENABLE_PIN) && !defined(CPU_MAP_ATMEGA328P)
98 #error "USE_SPINDLE_DIR_AS_ENABLE_PIN so pode ser usado com o processador 328p"
99 #endif /final do #ifndef HOMING_CYCLE_0
100
101 // -----
102
103 #endif //fial do \ifndef config_h

```