UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

DIORGINES MATTOS MACHADO

DESENVOLVIMENTO DE UMA MÁQUINA CNC COM HARDWARE E SOFTWARE OPEN SOURCE

CRICIÚMA

2018

DIORGINES MATTOS MACHADO

DESENVOLVIMENTO DE UMA MÁQUINA CNC COM HARDWARE E SOFTWARE OPEN SOURCE

Projeto de Pesquisa do Trabalho de Conclusão de Curso em Ciência da Computação, da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof. Sérgio Coral

CRICIÚMA

2018

DIORGINES MATTOS MACHADO

DESENVOLVIMENTO DE UMA MÁQUINA CNC COM HARDWARE E SOFTWARE OPEN SOURCE

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Automação.

Criciúma, 27 de junho de 2018.

BANCA EXAMINADORA

Prof. Esp. Sérgio Coral - UNESC - Orientador

Prof. MSc. Paulo João Martins - UNESC

Prof. Esp. Ênio José Peruchi - UNESC

Dedico este trabalho a aqueles que buscam na ciência uma resposta para suas invenções.

AGRADECIMENTOS

Agradeço primeiro a Deus pela força, saúde e fé para seguir meu caminho, a minha esposa Mabel Benincá e as minha filhas Maria Laura Benincá Mattos e Lívia Benincá Mattos que são as grandes razões da minha vida onde delas tirei a motivação que as vezes não tinha para prosseguir, ao meu grande professor e amigo e também orientador deste projeto Sérgio Coral que me mostrou o caminho na realização deste trabalho.

.

“A imaginação é mais importante que a ciência, por que a ciência é limitada, ao passo que a imaginação abrange o mundo inteiro.”

Albert Einstein

RESUMO

Desenvolvimento de um protótipo, modelo industrial, de uma máquina com comando numérico computadorizado para operações de usinagem utilizando hardware de comunicação Arduino UNO e CNC Shield com firmware GRBL implementado conforme configuração dos pontos de controle para realização dos movimentos nos eixos lineares X, Y e Z adaptados por motores de passo Nema 17, com o auxílio da computação gráfica pelo software Sketchup onde determina as linhas de segmento e secção geométrica de cada eixo através da extensão Sketch UCAM, ferramenta que transforma desenho em Linguagem CNC baseado nos fundamentos do algoritmo de Bresenham para interpolação de movimento por hardware em arquivo padrão de linguagem G-CODE interpretado pelo software Universal G-Code Sender para a realizar operações de usinagem.

Palavras-chave: Máquina CNC, Arduino, Arduino UNO, CNC Shield, Grbl, Linguagem CNC, G-code, Sketchup e usinagem.

ABSTRACT

Development of a prototype, industrial model, of a computerized numerical control machine for machining operations using Arduino UNO and CNC Shield communication hardware with GRBL firmware implemented according to configuration of the control points for the realization of the X, Y and Z linear axis movements adapted by Nema 17 step motors with the help of computer graphics software by Sketchup where it determines the lines of segment and geometric section of each axis through the Sketch UCAM extension, tool that transforms drawing in CNC Language based on the fundamentals of the algorithm of Bresenham hardware motion interpolation in standard G-CODE language file interpreted by Universal G-Code Sender software to perform machining operations.

Key-words: CNC Machine, Arduino, Arduino UNO, CNC Shield, Grbl, CNC Language, G-code, Sketchup and machining.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CNC	*Comando Numérico Computadorizado*
CAD	*Computer Aided Design*
CAM	*Computer Aided Manufacturing*
3D	*Três Dimensões*
CC	*Corrente Contínua*
CA	*Corrente Alternada*
LPT	*Local Printer Terminal*
BIOS	*Basic Input Output System*
SSP	*Synchronous Serial Port*
DMA	*Direct Memory Access*
MDI	*Manual Data Input*
ISO	*International Organization for Standardization*
NBR	*Norma Brasileira*
ASCII	*American Standard Code Information Interchange*
EEPROM	*Electrically Erasable Programmable Read Only Memory*
RISC	*Reduced Instruction Set computer*
SRAM	*Static Random Access Memory*
USART	*Universal Synchronous Asynchronous Receiver Transmitter*
SPI	*Serial Peripheral Interface*
AD	*Analógico Digital*

SUMÁRIO

1 INTRODUÇÃO

Desenvolver uma máquina com comando numérico computadorizado com plataforma de hardware de fácil acesso e softwares gratuitos com programação livre era uma tarefa difícil, hoje a automação está participando das pequenas oficinas e laboratórios com recursos e orientações que se traz através de sites de cientistas motivados a inovação, trocando seu espaço de conhecimento para toda rede interessada a invenções.

A indústria vive nos dias atuais “a era das máquinas livres” incentiva a adesão de novos inventores que revoluciona o uso de plataformas open source, que neste protótipo, os hardwares trazem vida em movimento sincronizado seccionado por eixos obedecidos por comandos através do software de programação (Goodrich, 2012).

O Arduino é um hardware que vem transformando o mercado do conhecimento, com bibliotecas já instaladas e com funções eletrônicas programadas através de uma IDE, pode-se gerar pulsos de movimentos elétricos e até mecânicos, com objetivo de auxílio ou transformação como por exemplo serviços de usinagem, para Pearce a plataforma Arduino torna simples a tarefa em realizar processos automatizados (Pearce, 2012).

O presente trabalho apresenta um estudo sobre o uso do Arduino em operações de usinagem com um sistema programado adaptado sobre uma máquina, detalhando a teoria do projeto dividido em pontos de construção mecânica, elétrica, eletrônica e configuração de hardware e software com metodologia utilizada e por fim uma análise dos resultados obtidos do protótipo.

1.1 OBJETIVO GERAL

Desenvolver uma máquina com fresa adaptada e com linguagem de comando numérico computadorizado utilizando comunicação entre hardwares e softwares open source (livres), para realização de operação em usinagem baseados nos estudos do algoritmo de Bresenham e coordenadas que resulta em movimentos entre eixos.

1.2 OBJETIVOS ESPECÍFICOS

a) estudar o algoritmo de Bresenham;

b) criar um programa CNC baseado no algoritmo de Bresenham;

c) encontrar hardware e softwares para comunicação entre computador e máquina;

d) identificar elementos elétricos e eletrônicos para automação dos movimentos dos eixos;

e) desenvolver e projetar os elementos mecânicos das peças que compõe o protótipo.

1.3 JUSTIFICATIVA

Este projeto se justifica pela relevância em desenvolver um protótipo através de novas tecnologias baseadas em linguagem de comunicação e pesquisas de elementos que contribui aos processos automatizados para usinagem.

Na indústria moderna a metodologia de produção acontece a meio da influência de ferramentas utilizadas por sistemas computadorizados, que realiza através da computação gráfica e manufatura assistida onde evidência e determina o fluxo de trabalho seriado para a transformação.

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 máquina CNC.

1.4 ESTRUTURA DO TRABALHO

Este projeto será apresentado em nove capítulos, sendo o capítulo um uma introdução do trabalho desenvolvido, com apresentação dos objetivos e os principais assuntos que estão detalhados no decorrer do trabalho.

O capítulo dois define os elementos eletromecânicos automatizado em um sistema básico de uma máquina com CNC.

O capítulo três traz uma maneira fácil de aprender as posições e sistemas de coordenadas.

O capítulo quatro trata a principal teoria do Algoritmo de Bresenham.

Capítulo cinco determina os principais softwares e hardwares para aplicação do equipamento em estudo.

O capítulo seis encontra-se os trabalhos correlatos utilizados como objeto de estudo e motivação para realização do trabalho.

No capítulo sete demonstra a construção mecânica, elétrica e eletrônica com todos os elementos montados e ajustados para o funcionamento do protótipo.

Capítulo oito relata passo a passo os softwares para uso de movimento axial e linear entre x,y e z com as configurações espaciais da área útil do protótipo.

Caítulo nove representa a programação cnc em base dos estudos no algoritmo de Bresenham.

No capítulo dez expressa a opinião com os resultados obtidos.

2 AUTOMAÇÃO

Historicamente o processo de automatização, era feito por meio da mecanização de tarefas que necessitavam esforços repetitivos, ou para compensar trabalhos árduos. A invenção da roda é um exemplo de automatização que facilitava trabalho de transporte. O foco era sempre simplificar o trabalho do homem, reduzindo o esforço físico (SILVEIRA, 1998).

Em 1960, a automação é sugerida para processos que recebem controle computacional. Automação é qualquer conjunto de sistemas, sustentado por processos computacionais, onde o objetivo é substituir o trabalho humano, em beneficio a qualidade dos produtos fabricados, redução das perdas de matérias primas e segurança das pessoas, com finalidade em diversas áreas, a automação surge para facilitar várias atividades humanas. Nas áreas residenciais possuem sistemas bem conhecidos como, máquinas de lavar louças, portões eletrônicos, micro-ondas, entre outros (MORAES, 2007).

Áreas públicas, e ou empresariais, caixas eletrônicos, controle de metrôs, semáforos e controle de pontos. E principalmente nas áreas industriais, sistemas automáticos de transporte, robótica, controle de qualidade, sistemas de segurança, processos de produção, entre outros (MARTINS, 2007).

2.1 ACIONAMENTOS EM SISTEMAS AUTOMATIZADOS

O motor possui em seu sistema de partida, um adaptador de controle para movimentação, estes sistemas são utilizados para alimentar elementos elétricos e eletrônicos de uma máquina ou equipamento que requer algum tipo de controlador, são aplicados em motores de corrente contínua (CC), corrente de fluxo contínuo movimentado por elétrons na mesma direção, ou corrente alternados (CA), gera um sinal mecânico, e com polia ou correia, se adapta uma velocidade de inércia, levando movimentos a outros elementos mecânicos, fazendo existir os movimentos de trabalho de uma máquina inteira ordenada através de um sistema eletrônico que recebe o controle operacional de trabalho (PAZOS, 2002).

Em muitas aplicações, têm-se como exigências de acionamento de avanço com precisão de posicionamento, homogeneidade e constância de velocidade, baixa inércia, resposta dinâmica, alta capacidade de sobrecarga, movimentos e paradas

rápidas com precisão, e reversão de movimentos. (PAZOS, 2002).

Os motores de indução são os motores mais utilizados nas indústrias, os motores síncronos possuem velocidade constante conforme sua frequência, os pólos do rotor seguem o campo imposto ao extrator de alimentação trifásica, as máquinas de corrente contínua, em função do seu princípio de funcionamento, permitem variar a velocidade de zero até a velocidade nominal ajustada, esta característica é de importância, desta forma é possível acionar em diversas aplicações que exige uma faixa variável de velocidade precisa, como acontece em máquinas operatriz CNC (PAZOS, 2002).

Servomotores são motores aplicados em servo-acionamentos, os circuitos de alimentação encontram-se em uma unidade chamada servo-conversor, este tipo de acionamento pertence ao sistema eletromecânico do controle de precisão, estes motores utilizados em acionamento podem ser corrente alternada ou corrente contínua, sabendo que servos em CA tem precisão menor que em CC, o motor de passo, devido a sua dimensão e seu custo está sendo muito mais aplicado do que os servomotores (PAZOS, 2002).

Figura 1 - Motor de Passo

Fonte: Pazos (2002)

Os motores de passo aplicam-se em máquina de baixo custo como as impressoras de tinta, sendo que existem variados modelos de motores de passo que estão à disposição do mercado e pode ser utilizado para diversos recursos, como por exemplo, mover robôs, circuitos de *cftv* e brinquedos, sua utilização é ampla (PAZOS, 2002).

Antes de aplicar um motor de passo, tem que conhecer algumas

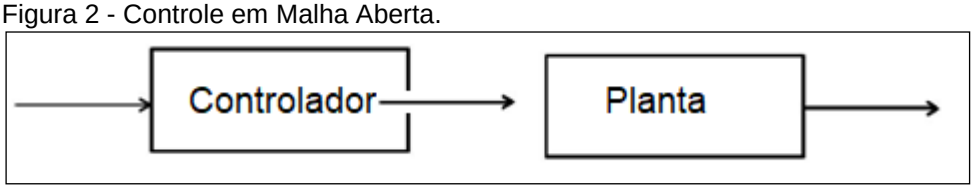
especificações técnicas para seu funcionamento, entender a tensão que alimenta, que tipo de corrente elétrica é utilizada nas bobinas e o ângulo de torque (PAZOS, 2002).

O torque é dado pela frequência aplicada, quanto maior a frequência na alimentação, menor é o torque, isso deve ao rotor, imã que roda na parte móvel do motor, onde tem menos tempo para se mover entre um ângulo ao outro, tendo isto se sabe que uma das características mais importantes é o ângulo, os mais comuns tem precisão de 0.72 graus, 1.8 graus, 3.6 graus, 7.5 graus, 15 graus e até 90 graus ou passos por volta de 500, 200, 100, 48, 24 e 4 (PAZOS, 2002).

O motor de passo possui diversas características, facilidade em comunicar-se com lógica digital permitindo precisão em seu deslocamento e velocidade de giro, pulso bloqueio com menor desgaste (PAZOS, 2002).

A ligação elétrica dos motores de passos divide-se em sistemas de controle em malha aberta, como mostra na Figura 2, onde o sistema digital controla os acionamentos dos atuadores na qual estiver programado, assim demonstra mais economia que a malha fechada, onde o sistema digital não conhece o recebimento das ações dos atuadores por parte do ambiente (PAZOS, 2002).

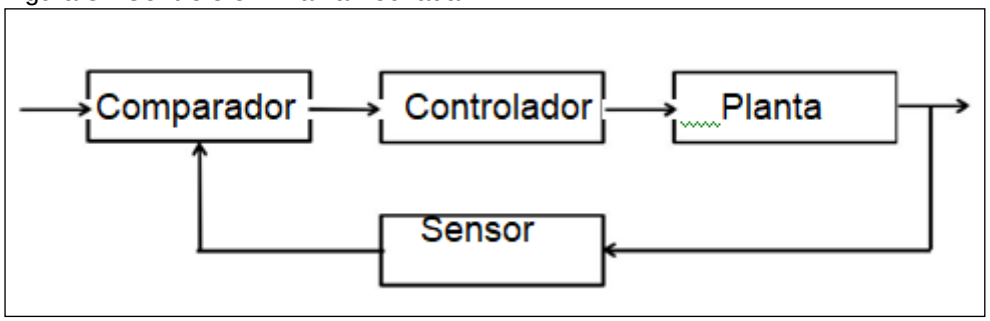
Figura 2 - Controle em Malha Aberta.



Fonte: Pazos(2002).

Os sistemas de controle em malha fechada, Figura 3, a unidade aciona os atuadores de acordo com os dados enviados aos sensores, assim monitorando o sistema, e qualquer alteração no sistema o controle permite o monitoramento em tempo real pelos sensores, sendo seguro e eficiente, mas o custo é uma desvantagem, este processo encarece a automação (PAZOS, 2002).

Figura 3 - Controle em Malha Fechada



Fonte: Pazos (2002).

2.2 COMUNICAÇÃO VIA PORTA PARALELA

Existem sistemas que comunicam com o computador por meio exterior com auxílio de uma porta paralela que controla alguns processos, a porta paralela é uma interface que pode se comunicar com oito bits simultaneamente, transmitindo cada bit por condução separado, este tipo de transmissão é indicado para comunicações em distâncias menores, a interferência depende do tamanho do cabo, quanto menor ou mais curto o cabo menor é a interferência e maior é a velocidade (BRAGA, 2005).

Figura 4 - Porta Paralela

Fonte: Braga (2005).

2.3 LINGUAGEM COMANDO NUMÉRICO COMPUTADORIZADO

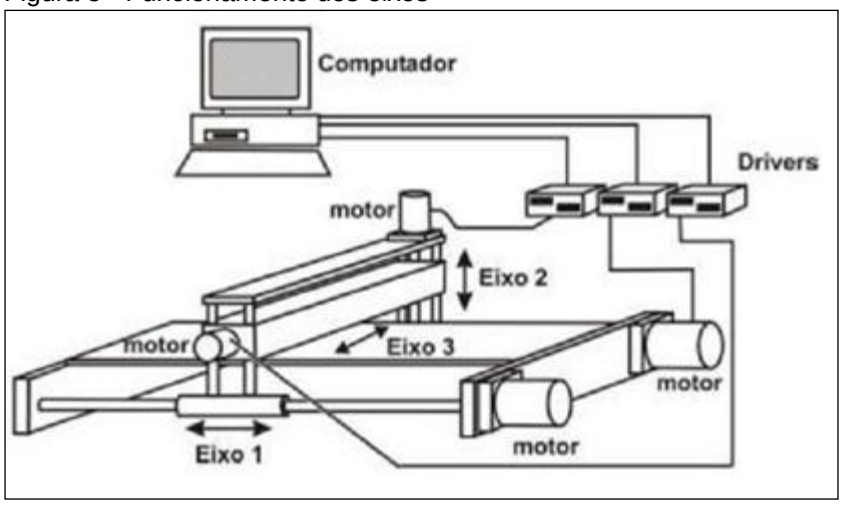
As máquinas com comando numérico computadorizado possuem um sistema eletrônico que recebe informações e transmite em formato de comando, compila esta informação em formato operacional e realiza aquilo que foi programado (MACHADO, 1990).

Um dos processos onde o CNC é aplicado é no setor mecânico, em principal nas áreas de cortes em material metálico, onde envolve esforço máquina e precisão, onde é suprido facilmente pelos motores de passo (FILHO, 2007).

Os eixos X e Y servem para movimentar a mesa principal, acionados por motor de passo em malha aberta por servo mecanismo com motores de corrente contínua ou acionadas por motores de indução em malha fechada que neste caso, a uma necessidade de sensores que fornecem o ângulo de posição e velocidade no eixo do motor (FILHO, 2007).

As máquinas CNC possuem normalmente três eixos de coordenação, onde se movimentam em linear e cruza os eixos entre si, eixo X responsável pelo movimento longitudinal, eixo Y pelo movimento transversal e eixo Z que define o movimento vertical ou altura onde a peça será cortada, (figura 5) (BATTI, 2007).

Figura 5 - Funcionamento dos eixos



Fonte: Batti (2007).

Os sentidos de movimento positivo e negativo precisam ser observado em cada eixo, algumas máquinas possuem coordenadas absolutas, situam-se sempre a um ponto fixo da peça, ou ponto zero, e as coordenadas incrementais usa de base a última posição (BATTI, 2007).

A tecnologia CNC nasceu no processo de usinagem, a grande maioria das peças no mercado são produzidas através deste processo com estas máquinas, até por processo de moldagem plástica, o molde é uma ferramenta de precisão feita por máquinas operadora CNC, sendo quanto maior o grau de precisão no acabamento maior é a perfeição, onde consiste em maior custo (PENTEADO, 2002).

Um dos mais antigos métodos utilizados pela industrialização é o processo de usinagem onde depende de alguns detalhes característicos, tipo de material a ser usinado, ferramenta de corte utilizado no processo, fluídos e metrologia aplicada, não apenas a máquina que evoluiu, mas também todos os detalhes que pertencem aos processos de usinagem, em principal as dimensões de precisão e a eletrônica de comunicação com a máquina (SILVA, 2006).

2.4 PROGRAMAÇÃO CNC

Os movimentos fazem parte de uma máquina CNC, movimentos lineares ou rotação, cada movimento é parte de um eixo associando em letras X, Y e Z (PEREIRA, 2006).

A programação pode ser de forma manual ou automatizado, classificada especificamente em programação manual ou programação em CAD e CAM (PEREIRA, 2006).

A programação manual é indicada a trabalhos de fresamento, a programação é feita sem utilização de recursos computacionais, são documentadas em manuscrito do programa, que são dados de listagem de posições da ferramenta em relação à peça a ser usinada, onde a máquina precisa seguir para executar o processo de transformação (PEREIRA, 2006).

A programação CAD e CAM são softwares onde o sistema CAD é um software gráfico computadorizado para criar uma peça graficamente, e o CAM é um sistema onde o software faz a leitura de cada margem do desenho em esboço e transforma em linha de programa, este processo de programação é o mais comum nas indústrias de usinagem de peças seriadas (PEREIRA, 2006).

A criação de um programa CNC por CAD e CAM passa pelas etapas de edição, simulação e transmissão do código para a máquina para ser compilado, onde o editor obtém de ferramentas gráficas, intervindo a erros no processo pelo motivo de ver a peça dentro das suas dimensões saber quais funções será realizado para sua criação, a simulação baseia-se na representação gráfica da trajetória das ferramentas em um plano escolhido, a transferência do código para a máquina tem como objetivo a redução do tempo, quando comparado à introdução via teclado da máquina, e a eliminação de eventuais erros de digitação, a mesma é feita pela porta serial RS232. Os parâmetros de comunicação, tais como velocidade, paridade, tamanho de palavra,

bits de parada, modo de transferência e o número da porta serial devem ser previamente definidos pelo usuário e o arquivo a ser transmitido deve conter apenas caracteres ASCII e possuir tamanho compatível com a memória disponível na máquina (PEREIRA, 2006).

A tecnologia de baixo custo proposta foi implementada com o intuito de reduzir o tempo de programação e, reduzir o custo dos processos de usinagem em ambientes fabris. A avaliação foi realizada em três etapas: seleção de empresas que atendessem a um determinado perfil, treinamento e avaliação de técnicos dessas empresas e teste do protótipo no chão de fábrica. Este sistema, embora não possa ser considerado um sistema CAD e CAM (o usuário digita as coordenadas para gerar a trajetória da ferramenta), quando comparado àqueles observados em empresas que ainda fazem uso da programação manual, representa uma alternativa viável para redução do tempo total de usinagem e consequentemente de custos de fabricação. (PEREIRA, 2006)

3 SISTEMA DE COORDENADAS

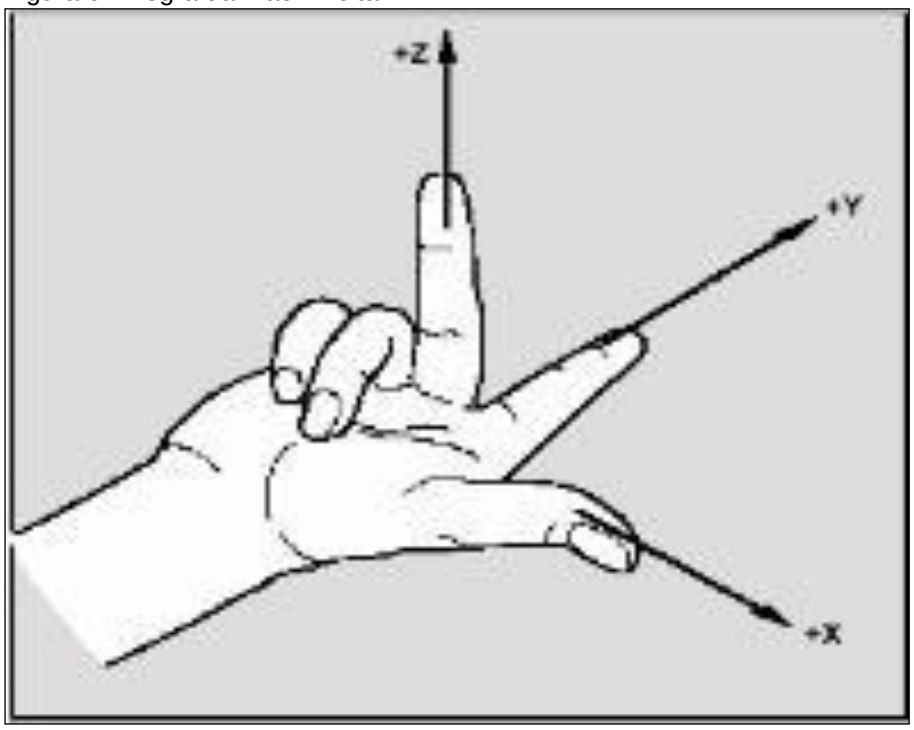
As máquinas com comando numérico computadorizado possuem um comando baseado em plano cartesiano chamada de coordenadas, onde é utilizada para coordenar as medidas de movimento dos eixos para construir ou usinar determinado material até alcançar um perfil geométrico (SILVA, 2014).

Para que a máquina possa trabalhar com as posições especificadas, estas têm que ser declaradas em um sistema de referência, que corresponde aos sentidos dos movimentos dos carros (eixos X, Y, Z) (SILVA, 2014).

O sistema de coordenadas da máquina é formado por todos os eixos existentes fisicamente na máquina (SILVA, 2014).

As direções dos eixos seguem a regra da mão direita, conforme Figura 6, deve pensar que o programa sempre segue a trajetória da ferramenta (SILVA, 2014).

Figura 6 - Regra da Mão Direita



Fonte: Silva (2014).

3.1 COORDENADAS ABSOLUTAS

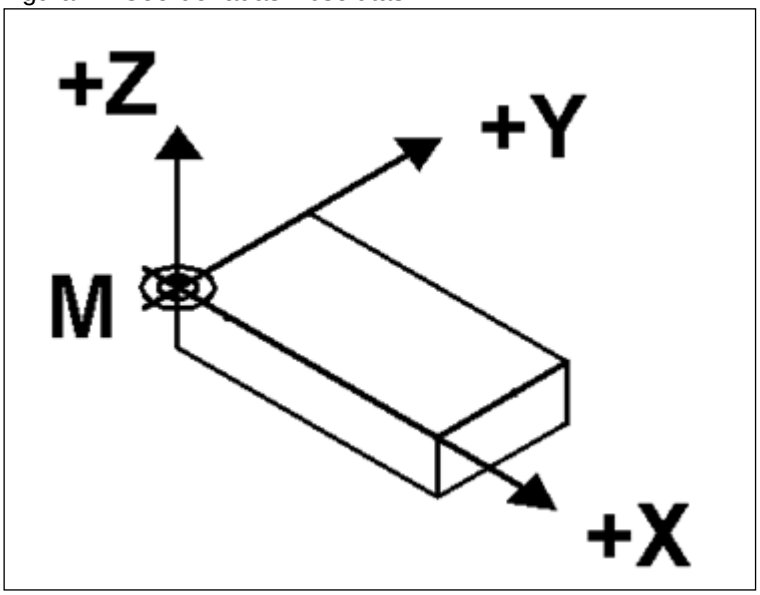
No modo de programação em absoluto as posições são medidas da posição zero atual (zero peça) estabelecido. Com vista ao movimento da ferramenta isto significa a dimensão absoluta descreve a posição para a qual a ferramenta deve ir (SILVA, 2014).

3.1.1 Função G90 - Coordenada Absoluta

A coordenada absoluta como mostra na Figura 7, são definidas através do código G90 e seus valores sempre deverão estar em relação ao ponto zero da peça (SILVA, 2014).

Eixo X refere-se às medidas na direção longitudinal da mesa, eixo Y refere-se às medidas na direção transversal da mesa e eixo Z refere-se às medidas na direção vertical da ferramenta (SILVA, 2014).

Figura 7 - Coordenadas Absolutas



Fonte: Silva (2014).

3.2 COORDENADAS INCREMENTAIS

No modo de programação em incremental as posições dos eixos são medidas a partir da posição anteriormente estabelecida. Com vista ao movimento da ferramenta isto significa a dimensão incremental descreve a distância a ser percorrida pela ferramenta a partir da posição atual da mesma (após o último movimento) (SILVA, 2014).

3.2.1 Função G91 – Coordenada Incremental

Coordenadas incrementais são definidas através do código G91 e seus valores sempre serão obtidos em relação ao último posicionamento da ferramenta (SILVA, 2014).

4 ALGORITMO DE BRESENHAM

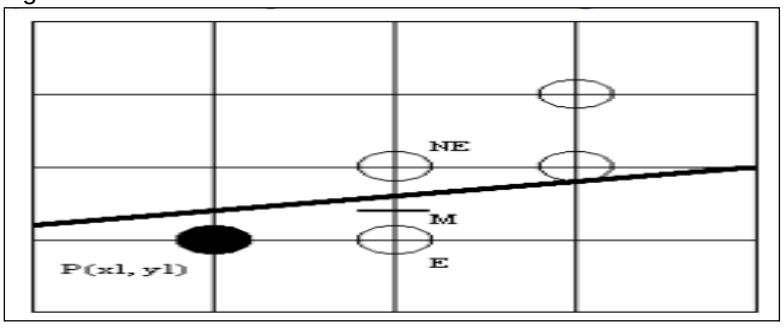
É relativamente fácil criar um algoritmo para traçar retas e gerar pontos aproximados sobre bases matemáticas de pontos flutuantes. Hoje possivelmente as ULA’s dos computadores possuem instruções para trabalhar com este tipo de sistema flutuante de forma rápida, porém no caso de microcontroladores esta situação se

converte em problema devido à baixa velocidade de operação e as reduzidas instruções (MORO, 2009).

O uso dos microcontroladores na automação é um problema se não usados corretamente. Usar ponto flutuante exige do processador vários ciclos de clock acarretando na defasagem da comparação e controle no processo automatizado. Para contornar esta situação opta-se por algoritmos de alto nível minimizando o trabalho computacional. Em 1965 Jack E. Bresenham, então funcionário da IBM propõe um algoritmo eficaz para a época que calculava melhor ponto nas trajetórias retilíneas por meio de lógica com números inteiros (MORO, 2009).

Foi proposto por Bresenham em 1965 um algoritmo clássico que utiliza apenas variáveis inteiras, Algoritmo de Bresenham ou conhecida no mercado da computação gráfica como Algoritmo do Ponto-Médio, o algoritmo permite fazer os cálculos de (xi+1, yi+1) incrementalmente, usando os cálculos já feitos para (xi, yi), o algoritmo assume que a inclinação está entre 0 e 1 no 1º octante (outras inclinações podem ser usadas por simetria), e o ponto (x1, y1) é o inferior esquerdo, e (x2, y2) é o superior direito. Partindo do ponto P(x1, y1), a escolha do próximo ponto a ser desenhado é analisando o ponto médio M, sendo que, se M está acima da reta, o próximo ponto a ser desenhado é o da direita (E), e se M estiver abaixo da reta, o próximo a ser desenhado é o ponto acima e a direita (NE), (FERREIRA, 2004).

Figura 8 - Reta



Fonte: Ferreira (2004).

Utilizando esta função é possível analisar que P(x,y) é zero para pontos sobre da reta, negativa para pontos acima e positiva abaixo. Logo em seguida é analisado o ponto médio M com base ao pixel atual P e verificado qual pixel foi escolhido, ME ou M. Uma nova comparação lógica é determinada para saber o quanto e qual valor de x e y será incrementado para o próximo pixel dependendo do pixel atualmente escolhido (BRESENHAM, 1965, tradução nossa).

Bresenham (1965, tradução nossa) por meio dos cálculos e eliminando a fração da equação primitiva da reta encontra equações de decisões, ou métodos que satisfazem o coeficiente angular caso as seleções dos pixels sejam diferentes. As equações de decisões são consideradas constantes de Bresenham, e são elas:

$$D_{start} = 2\Delta y - \Delta x \qquad \text{condição } inicial$$
$$D_{new} = D_{old} + 2\Delta y \qquad se\ I\ for\ escolhido$$
$$D_{new} = D_{old} + 2(\Delta y - \Delta x) \qquad se\ S\ for\ escolhido$$

Por fim o Bresenham (1965, tradução nossa) apresenta condições ainda de recursão, caso a inclinação da reta seja invertida, x1 > x2, para satisfazer o algoritmo por completo. O algoritmo de Bresenham pode ser desenvolvido em qualquer linguagem, Figura 9, e em qualquer sistema de modo eficaz, por possuir lógica inteira.

Figura 9 – Algoritmo de Bresenham em C

```
/** Desenha uma reta entre dois pontos **/
void DrawLine(point p0, point p1, color color){
        int slope;
        int dx, dy, incE, incNE, d, x, y;

        // Onde inverte a linha x1 > x2
        if (p0.x > p1.x){
            DrawLine(p1, p0, color);
            return;
        }

        dx = p1.x - p0.x;
        dy = p1.y - p0.y;

        slope = (dy < 0) ? -1 : 1;
        if (dy < 0){
            dy = -dy;
        }
```

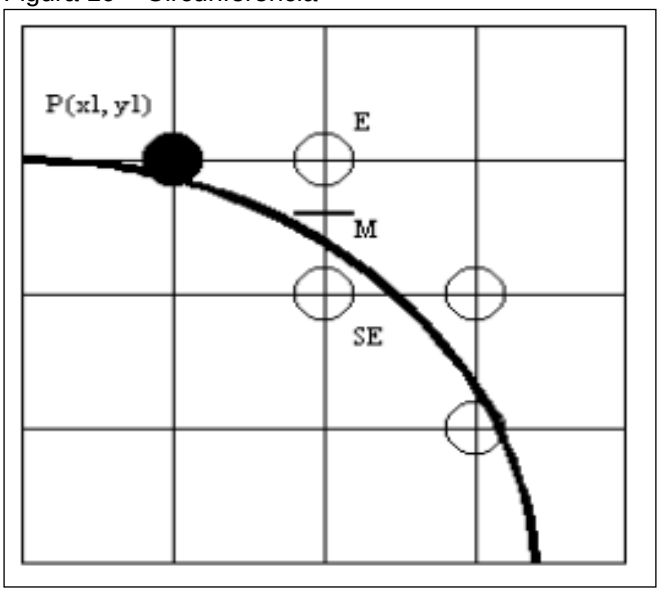
```
        // Constante de Bresenham
        incE = 2 * dy;
        incNE = 2 * dy - 2 * dx;

        d = 2 * dy - dx;
        y = p0.y;

        for (x = p0.x; x <= p1.x; x++){
            PutPixel(x, y, color);
            d += (d <= 0)? incE: incNE;
            if(d > 0) y += slope;
        }
}
```

Fonte: Dalmolin (2014).

Para construção de circunferências utiliza-se o Algoritmo do Ponto-Médio para Circunferências, uma equação com centro na origem e raio R, no plano cartesiano é dada por: $x^2 + y^2 = R^2$, seja a função $F(x, y) = x^2 + y^2 - R^2 = 0$, o valor 0 é sobre a circunferência, positivo fora dela e negativo dentro, considerando um arco de 45º, onde o ponto inicial é dado por P(x, y) e sendo (x = 0) e (y = R), a escolha do próximo ponto a ser desenhado é analisando o ponto médio M, sendo que, se M está

dentro da circunferência, o ponto escolhido é o próximo a direita (E), e se M está fora ou sobre a circunferência, o ponto escolhido é o próximo a direita e abaixo (SE), (FERREIRA, 2004).

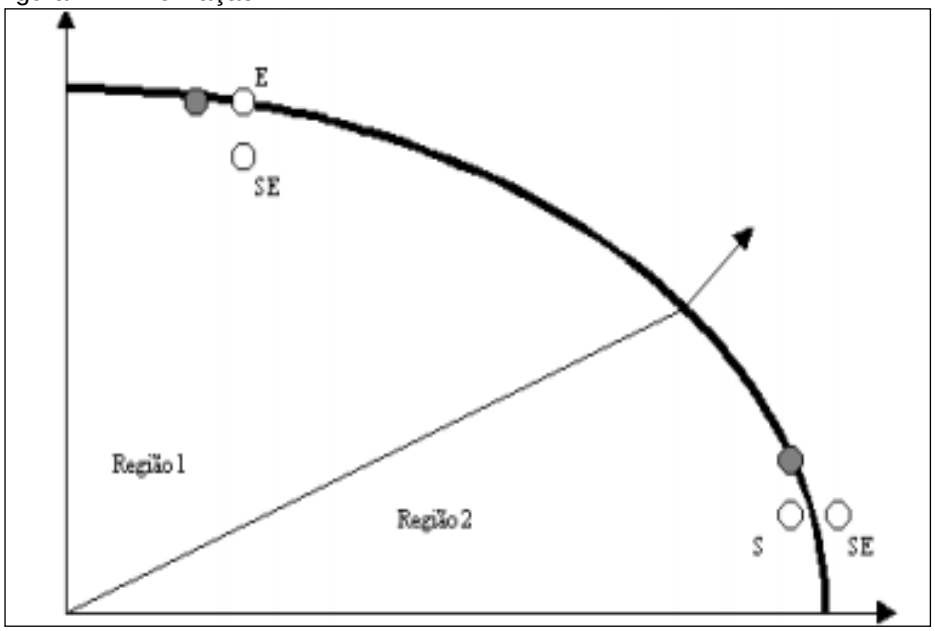
Figura 10 – Circunferência



Fonte: Ferreira (2004).

Para construção de elipses, utiliza-se Algoritmo do Ponto-Médio para Elipses, a equação da elipse centrada em (0, 0) é dada por: $F(x, y) = b^2x^2 + a^2y^2 - a^2b^2 = 0$, onde 2a é o comprimento do eixo maior (eixo x) e 2b é o comprimento do eixo menor (eixo y), como no traçado da elipse ocorre mudança na inclinação, divide-se o 1º quadrante em duas regiões, sendo, o limite entre as duas regiões é o ponto da curva cuja tangente tem inclinação igual a –1, assim, enquanto $a^2y > b^2x$ permanece-se na região 1, caso contrário, muda-se para a região 2 (FERREIRA, 2004).

Figura 11 – Inclinação



Fonte: Ferreita (2004).

Seja a função $F(x, y) = b^2x^2 + a^2y^2 - a^2b^2 = 0$, o valor 0 é sobre a elipse, positivo fora dela e negativo dentro, considerando um arco de 45º, onde o ponto inicial é dado por P(x, y) e sendo (x = 0) e (y = b), a escolha do próximo ponto a ser desenhado é analisando o ponto médio M, na região 1, se M está dentro da elipse, o ponto escolhido é o próximo a direita (E), e se M está fora ou sobre a elipse, o ponto escolhido é o próximo a direita e abaixo (SE), na região 2, se M está dentro da elipse, escolhe-se (SE), e se M está fora ou sobre a elipse, escolhe-se (S), como os incrementos são calculados para o 1º e 2º octantes, deve-se usar simetria de ordem 4 para desenhar os pontos dos demais octantes, formando a elipse (FERREIRA, 2004).

Em um sistema automatizado a última área consiste no software gerenciador de todo sistema. Este software entra como analisador das informações geradas por todo processo automatizado, por meio de comunicações entre o sistema de controle. O Arduino por possuir maior abstração facilita no processo de comunicação entre um computador e o sistema. Esta comunicação deve ser elaborada para receber os dados de um software e converte-los em passos para os motores. Protocolos devem garantir à exatidão no recebimento das informações enviadas a máquina, de tal maneira que comandos e dados sejam interpretados pelo Arduino (BURIGO, 2014).

5 COMUNICAÇÃO

A troca de informações entre dois dispositivos através de um meio de comunicação é conhecida como comunicação de dados. Os dispositivos precisam ser parte de um sistema feito com a união de hardware e software (FOROUZAN, 2006).

Forouzan (2006) destaca três características básicas, que garantem a eficiência e o bom funcionamento da comunicação:

a) entrega: deve garantir o recebimento dos dados corretos ao dispositivo de destino;

b) confiabilidade: o sistema deve garantir a entrega dos dados;

c) tempo de atraso: envio dos dados possui um tempo limite, necessitando ser o mais eficiente possível.

Tanenbaum (2002) classifica a comunicação de acordo com o fluxo de dados a serem trafegados:

a) simplex: somente um dispositivo é capaz de transmitir e o outro apenas é capaz de receber, comunicação é unidirecional. Rádios são exemplos desta comunicação*;*

b) *half-duplex*: ambos dispositivos recebem e enviam dados, mas não ao mesmo tempo. Quando um transmite o outro recebe e vice-versa. Um exemplo são os radiocomunicadores usados por seguranças e militares;

c) *full-duplex*: este fluxo descreve o compartilhamento da capacidade do meio de transmissão, ou seja, os dispositivos podem enviar e receber dados simultaneamente. Um exemplo de *full-duplex* são os celulares.

Um sistema de comunicação básico, Tanenbaum (2002), possui no mínimo cinco componentes:

a) mensagem: dados que serão transmitidos;

b) transmissor: dispositivo usado para transmitir a mensagem;

c) receptor: dispositivo usado para receber a mensagem;

d) meio: caminho físico por onde trafega a mensagem, do transmissor para o receptor;

e) protocolo: regras que padronizam o envio e o recebimento das informações.

5.1 GCODE

GCODE é uma linguagem representada graficamente a um programa funcional, sendo concebida para ser de fácil entendimento para programar e acoplar em máquinas, assim podendo gerar testes de entrada e as saídas facilmente interpretáveis (AXFORD, 1991).

Esta representação mais gráfica de linguagem foi inicialmente concebida por motivos de ensino. A representação de um programa funcional podia então ser impressa no formato ASCII, que é o formato GCODE. Este formato podia ser facilmente transmitido de uma máquina para outra usando e-mail ou outra forma de transferência de dados (AXFORD, 1991).

O GCODE hoje é amplamente utilizado em máquinas de automação, como em máquinas de comando numérico computadorizado (CNC), que é o tipo de máquina que se trata neste projeto. Ou seja, o GCODE está sendo utilizado para enviar comandos para a máquina (AXFORD, 1991).

5.2 GRBL

O programa GRBL é um interpretador de GCODE e também serve como interface para transmitir os comandos a partir do computador. Ou seja, ele atua como o software que envia os comandos de GCODE para o Arduino e também como o interpretador de GCODE no próprio Arduino (BENGLER, 2014).

Como interpretador de GCODE no Arduino, o GRBL funciona muito bem em sistemas baseados em Atmega328, por mais que haja certa incompatibilidade com outros sistemas isto não foi um grande problema (BENGLER, 2014).

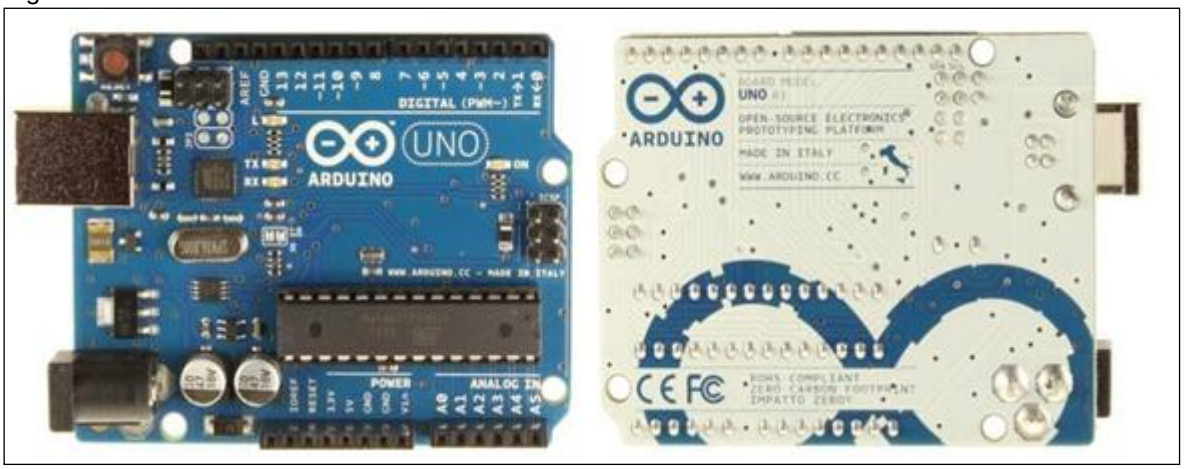
Como controlador ele tem uma interface gráfica para diversos sistemas operacionais que permite fazer upload de comandos escritos em GCODE de um arquivo (BENGLER, 2014).

5.3 ARDUINO

Arduino é uma plataforma de prototipação eletrônica *open-source* flexível que utiliza o microcontrolador ATMega328. O Arduino pode receber sinais de vários sensores eletrônicos e lidar com essas informações para controlar motores, luzes,

servos e qualquer outro atuador. O modelo utilizado no robô é o Arduino UNO (figura 12) (MONK, 2010).

Figura 12 - Arduino



Fonte: Monk (2010).

O microcontrolador utilizado no Arduino é o ATMega328 da Atmel, que utiliza a arquitetura de Harvard e é de 8 bits, RISC, com 32KB de memória flash, 1KB de EEPROM, 2KB de SRAM, 32 registradores de uso geral, e três temporizadores com contadores, uma USART, portas para comunicação SPI, seis conversores AD de 10bits e um watchdog timer, entre outras características. A tensão de operação dele é entre 1,8 e 5,5 V (MONK, 2010).

O software que controla os pinos e as ações do Arduino pode ser desenvolvido no programa chamado Arduino IDE, software gratuito, que conta com uma interface gráfica simples e intuitiva, e aceita códigos em C/C++ (MONK, 2010).

O código a ser embarcado ou programado, deve estar no formato aceito pelo próprio Arduino, e contar com as seguintes chamadas de procedimento, setup que ajusta as configurações das portas de entrada e saída e a comunicação serial, e loop que é um laço infinito onde o usuário cria a rotina do seu programa (MONK, 2010).

5.4 CAD / CAM

O sistema CAD é empregado nas atividades de desenvolvimento de produtos (design industrial) e projeto. O CAD tem por objetivo auxiliar no desenho e modelamento de peças pela interação com o computador em que são definidas todas as informações geométricas necessárias para a manufatura. (SOUZA, 2009)

Atualmente são desenvolvidos sistemas CAD para diversas aplicações de engenharia, arquitetura e design, existem também, sistemas CAD para cada sub-área de aplicação, como os módulos específicos para modelagem de chapas metálicas, de tubulações, circuitos elétricos, modelamento de formas complexas, desenhos 2D, entre outras aplicações (SOUZA, 2009).

A classificação dos sistemas CAD dá-se conforme sua possibilidade de aplicação, isto envolve fatores relacionados ao custo e à capacidade de representação geométrica destes sistemas, esta classificação auxilia as empresas ao adquirir o sistema mais eficiente para a aplicação desejada, esta classificação é dividida em três categorias, sistemas CAD de pequeno, médio e grande porte, onde os sistemas CAD de pequeno porte (low-end) correspondem aos softwares CAD utilizados para representar objetos e formas geométricas em duas dimensões tais como retas, círculos e curvas, para a representação gráfica, nos sistemas CAD de médio porte (middle-end), a principal característica é a representação geométrica em três dimensões, esta classe de software pode gerar objetos com informações superficiais e características mecânicas como centro de gravidade e volume (SOUZA, 2009).

Um sistema CAD tridimensional pode auxiliar em diferentes etapas de projeto, como na concepção, na análise dos componentes e montagens, centro de massa e gravidade entre outros, representando uma potente e indispensável ferramenta, e pode ser responsável por várias funções como o modelamento tridimensional, análise de formas geométricas para manufatura, análise de interferências em peças e conjuntos-montados, definição de volume e centro de massa do produto, geração de desenhos em duas dimensões com cotas e dimensionamentos, análise de suavidade de continuidade entre superfícies e, por fim, o gerenciamento do projeto (SOUZA, 2009).

Os sistemas CAD de grande porte (high-end) foram desenvolvidos e utilizados por grandes corporações, como a General Motors, IBM, Dassault, entre outros, estes sistemas são compostos por vários módulos dentro do mesmo software, os módulos podem envolver outros sistemas computacionais de análises e auxílio na concepção do projeto, esta categoria de sistema CAD engloba todas as características apresentadas até o momento (SOUZA, 2009).

6 TRABALHOS CORRELATOS

Para entendimento foram pesquisados trabalhos que se encaixam na área da educação, onde podemos entender as principais dificuldades dentro da realidade teórica e pratica do objeto que está sendo projetado para facilitar os estudos em comandos CNC.

6.1 EDUCAÇÃO EM CONTROLE E AUTOMAÇÃO EM AMBIENTE ADVERSO: ESTUDO DE CASO DE UMA EXPERIÊNCIA TUTORIAL

O presente trabalho de Hélvia Hortência Barcelos Carvalho da Universidade Federal de Juiz de Fora, Faculdade de Engenharia do Departamento de Energia em 2006, trata de um tema recorrente, de forma crescente, nos últimos anos: a educação em controle. É ponto consensual que a educação em controle deva fomentar as bases para um aprendizado contínuo que habilite o engenheiro de controle a lidar com os complexos, crescentes e emergentes problemas da área, dessa forma, ela deve permitir estabelecer e manter elevados padrões de excelência que possibilitem o aprendizado adequado de suas bases e conceitos fundamentais.

A proposta deste trabalho é efetuar uma análise dos princípios e diretrizes adotadas, bem como dos resultados obtidos, nestes 15 anos de existência do Programa de Educação Tutorial - PET Engenharia Elétrica enquanto espaço privilegiado para formação de profissionais de excelência, e as formas de inserção, neste trabalho, de uma educação estruturada em controle e automação (CARVALHO, 2006).

6.2 CONTROLE DE UM SISTEMA XY COM MOTORES DE PASSO POR MEIO DO ALGORITIMO DE BRESENHAM

Este trabalho de Israel Burigo Dalmolin, da Universidade do Extremo Sul Catarinense, apresentado na conclusão de curso em 2014, reporta a utilização de motores elétricos onde facilita no controle de produção e garante força e precisão na automatização, para controlar estes motores são necessários sistemas eletroeletrônicos que transformam dados em energia elétrica, o conjunto de controle e motores é inclusos em diversos maquinários para incontáveis finalidades, uma

destas finalidades é o controle de sistemas XY, máquinas que deslocam uma cabeça em coordenadas e podem cortar, plotar, bordar, entre outros.

Estes sistemas geralmente são privados e de difícil confecção, além de possuir controle por meio de lógicas flutuantes, tornando o processamento do deslocamento mais lento, acarretando em defasagem produtiva, para tanto, foi possível confeccionar circuitos eletrônicos viáveis que permitem controlar motores específicos em coordenadas. Além de aperfeiçoar o rendimento do processamento de tais circuitos, implementando algoritmos que executam lógica inteira.

Por sua vez, se fez necessário a elaboração da comunicação entre um determinado usuário e o sistema, estabelecendo a validação das ações do usuário, consequentemente, disponibilizando um sistema eficiente, de fácil aquisição e controle, porém existe a necessidade de implementar um software que compreenda está comunicação acarretando no auxílio de um indivíduo terceiro (DALMOLIN, 2014).

7 CONSTRUÇÃO PROTÓTIPO MÁQUINA CNC

Um protótipo CNC ao construir engloba em todo seu projeto uma instalação mecânica, elétrica e eletrônica, não se considera cálculos de complexidade sobre sua resistência estrutural, pelo motivo que se trata de um equipamento em menor escala, apenas considera o objetivo de complemento ao ensino de programação em máquinas automatizadas com sistema seriado.

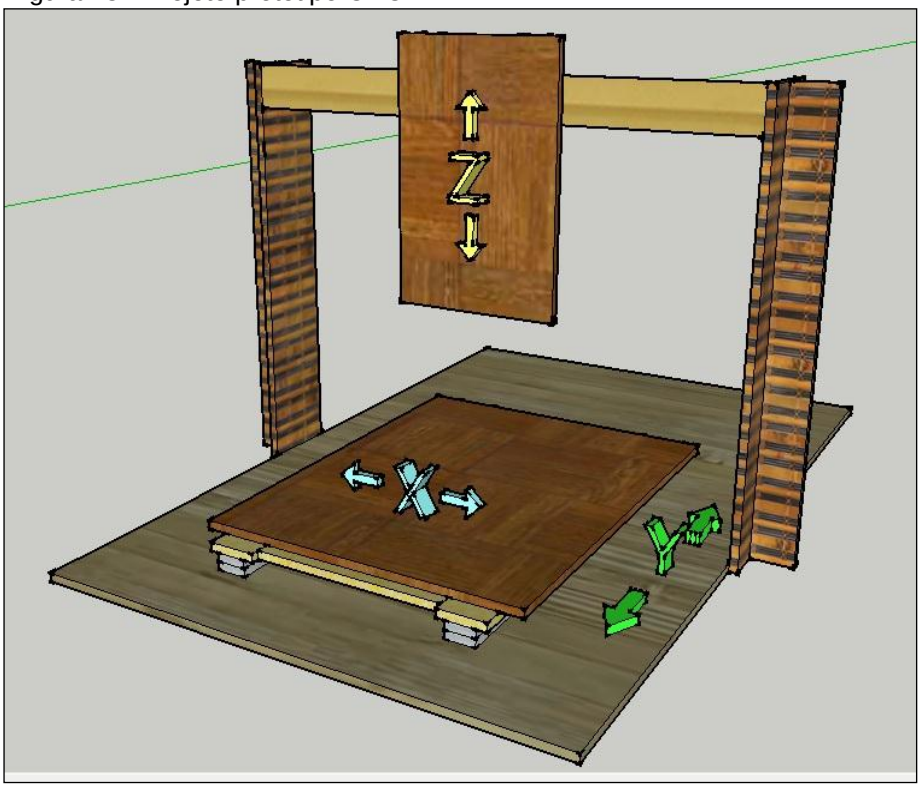
7.1 MECÂNICA

Para construir um protótipo máquina CNC no primeiro ponto determina-se sua área útil de trabalho e o tipo de material produzido ou usinado e o segundo ponto é a característica da base padrão dos eixos de qual forma será alojado, dentro destes requisitos o protótipo obtém estrutura mecânica e corpo visual de uma máquina CNC.

7.1.1 Projeto máquina estrutural

O protótipo da máquina CNC deste trabalho seguiu o modelo *Router*, que é muito utilizado em usinagem de materiais maleáveis como madeiras e plásticos, a máquina opera com três eixos direcionais, sabendo que o eixo X responde a direção longitudinal, eixo Y responde a direção transversal e eixo Z responde a direção vertical, estes eixos trabalham independente em suas posições de base (figura 13).

Figura 13 - Projeto protótipo CNC



Fonte: Do autor.

7.1.1.1 Elementos de máquina

Ao executar o projeto buscar investir em elementos de máquina de baixo custo, conforme representa no apêndice A os valores que se investiu por elemento, sendo chapas em madeira MDF, parafusos, arruelas, porcas, acoplamentos para motores de passo, barras roscadas e corrediças telescópicas, estes materiais são leves e de fácil aplicação também são resistentes a impactos gerados pelos movimentos de corte na usinagem (figura 14).

Figura 14 – Elementos de máquina

Fonte: Do autor.

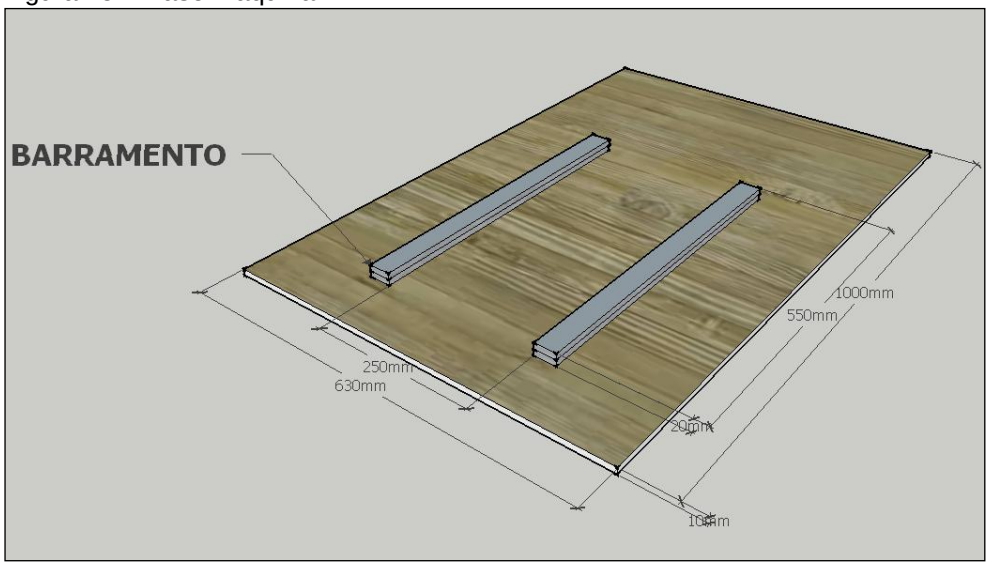
Toda estrutura dos eixos X, Y, Z e base de apoio do corpo da máquina é feita de chapa em madeira MDF.

7.1.1.2 Montagem máquina

Em qualquer máquina CNC encontra-se uma base de sustentação para cada eixo, onde se fixa os diversos membros ou seções, no protótipo seus eixos Y e X faz parte da mesa onde é colocada a peça bruta que será conformada pelo processo de usinagem, no eixo Z, a base sustenta a ferramenta de conformação, neste protótipo aplica-se uma micro retífica para função de eixo árvore ou ferramenta de corte.

Na base de sustentação se instala os barramentos que se utiliza no carro principal do eixo Y, conforme figura 15 que representa suas dimensões em milímetros.

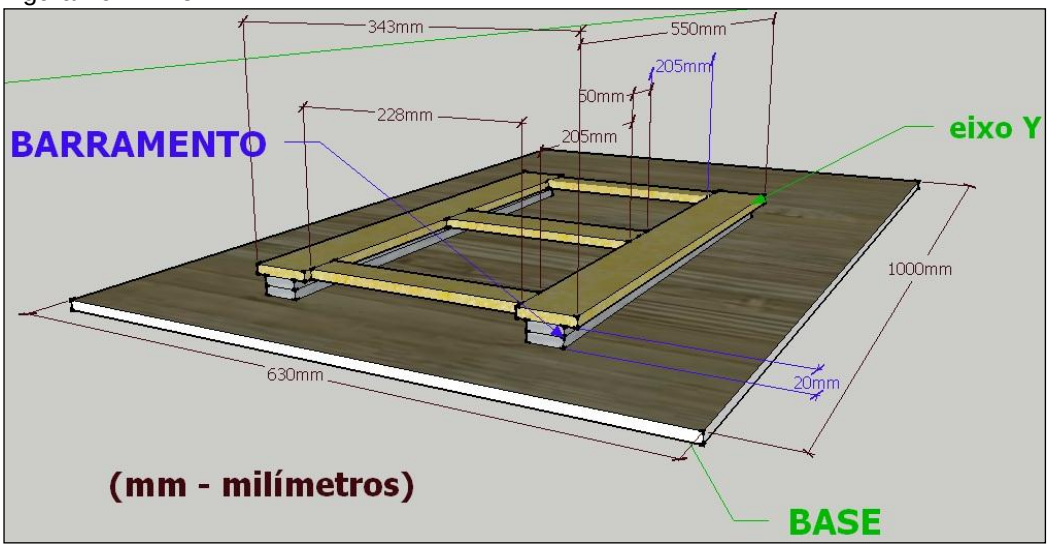
Figura 15 – Base Máquina



Fonte: Do autor.

No eixo Y, conforme padrão de instalação, o mesmo fixa-se sobre a área da base unida por um barramento deslizante ou rolamento de gaveta (corrediça para madeira), para realizar o movimento linear transversal, conforme figura 16 e suas principais medidas de construção.

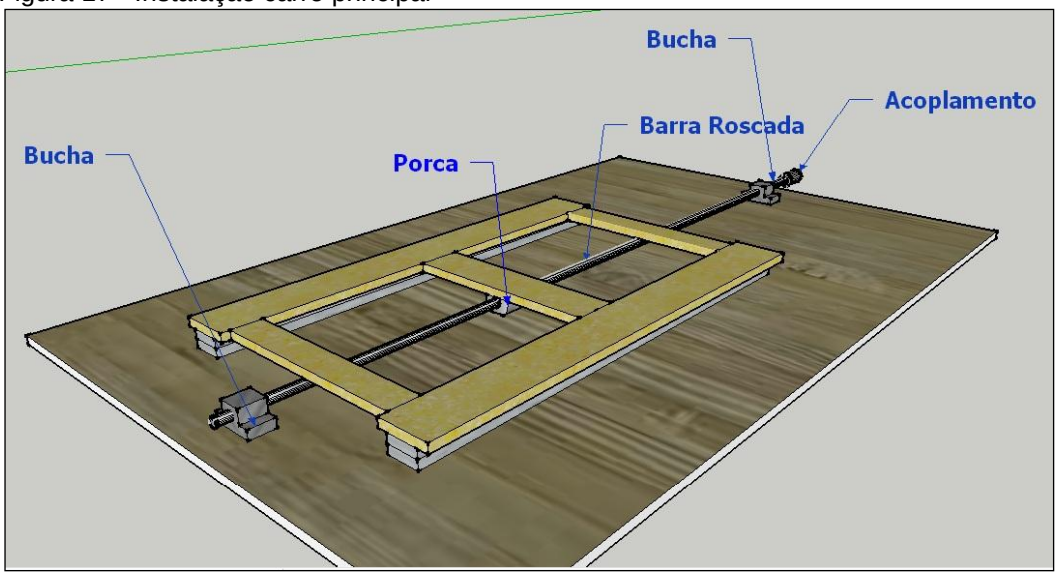
Figura 16 – Eixo Y



Fonte: Do autor.

Com uma porca fixa no centro de sua base unida a uma barra roscada sobre buchas presas nas extremidades com um acoplamento, firma-se a sustentação (figura 17).

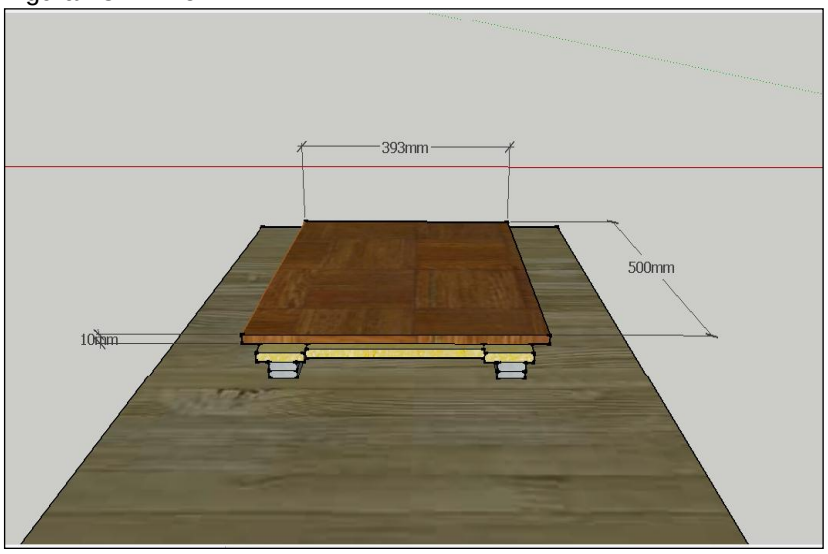
Figura 17 - Instalação carro principal



Fonte: Do autor.

O eixo X é responsável pelo movimento longitudinal, encontra-se instalado sobre o carro principal ou eixo Y, com barramentos deslizantes ou rolamento de gaveta (corrediços para madeira), conforme figura 18 e suas principais dimensões.

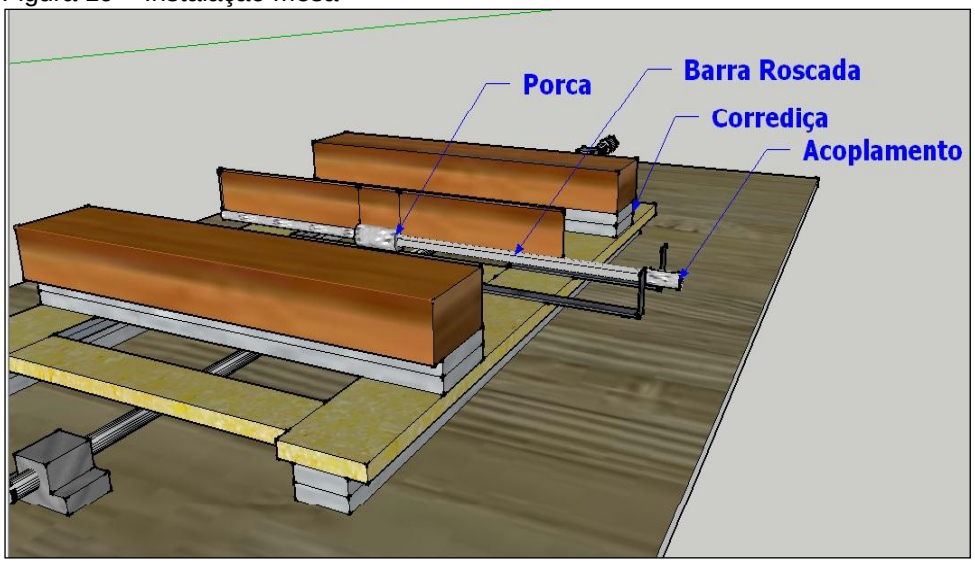
Figura 18 – Eixo X



Fonte: Do autor.

Para realizar o movimento linear longitudinal, se instala uma porca no centro da mesa de apoio ou mesa principal que está sobre dois barramentos deslizantes ou rolamento de gaveta (corrediços para madeira) e uma barra roscada com acoplamento preso na extremidade (figura 19).

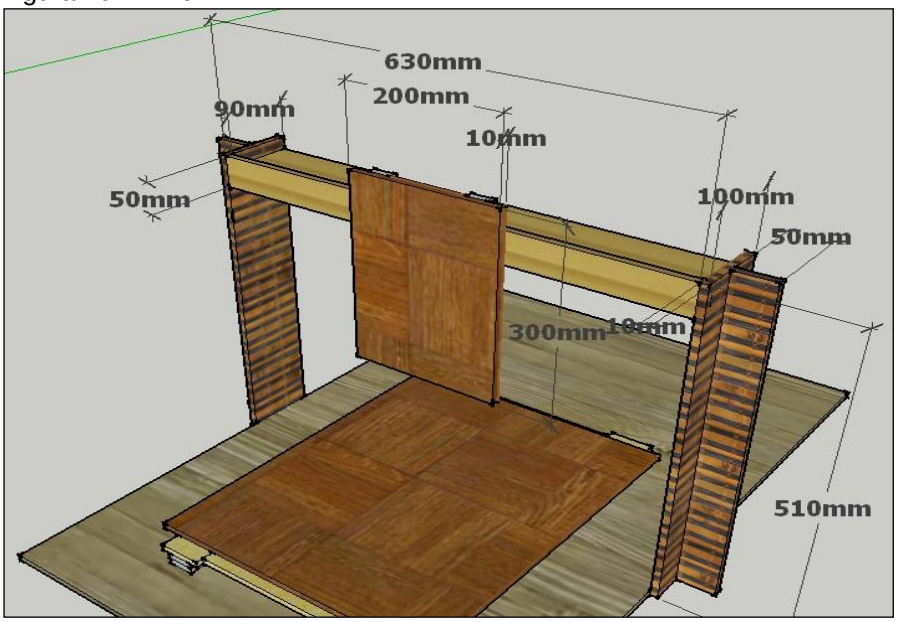
Figura 19 – Instalação mesa



Fonte: Do autor.

O eixo Z movimenta-se no sentido vertical onde é responsável pela furação ou corte superficial, fixa-se na plataforma vertical sobre os barramentos ou corrediças de rolamento conforme figura 20 e suas medidas.

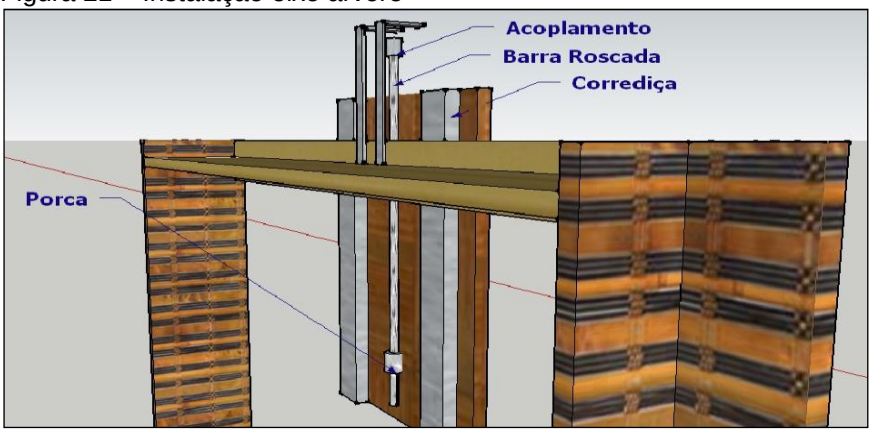
Figura 20 – Eixo Z



Fonte: Do autor.

Para ter avanço linear vertical fixa-se uma porca no centro da mesa do carro eixo árvore, sobre dois barramentos deslizantes ou rolamento de gaveta (corrediços para madeira) com uma barra roscada e em seu extremo um acoplamento, assim acontece o movimento de direção vertical ao iniciar o sentido de giro do motor (figura 21).

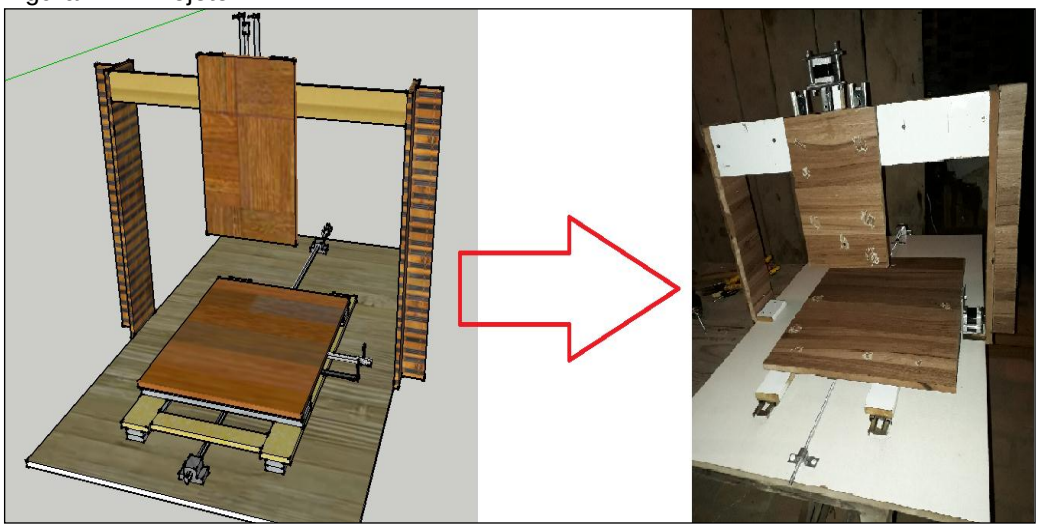
Figura 21 – Instalação eixo árvore



Fonte: Do autor.

No mundo dos projetos, o desenho técnico ajuda a tracejar uma máquina real, usa-se para definir sua forma interior, exterior e de trabalho, os desenhos foram feitos no software Sketchup Pro 2016, na figura 22 observa-se o resultado em esboço e projeto real.

Figura 22 – Projeto



Fonte: Do autor.

7.2 ELÉTRICA E ELETRÔNICA

Instala-se a eletroeletrônica de uma máquina CNC por partes, sendo módulos da placa controladora, módulo dos drivers de potência, módulo dos atuadores, fonte de energia e ferramenta elétrica ou eixo árvore.

7.2.1 Placas de controle

Uma placa de controle é responsável pela comunicação de dados da máquina, interpreta os dados a ser enviados pelo computador, a placa de controle coordena os movimentos executados pelos eixos sendo a aceleração e acionamento do motor de passo conforme desejável.

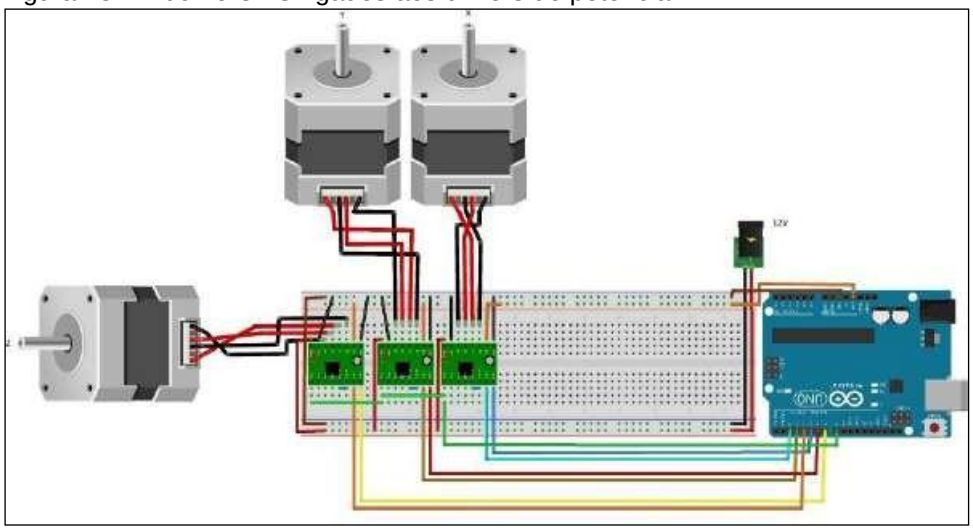
7.2.1.1 Arduino UNO

A placa Arduino UNO integra com as configurações de *firmware* da *Grbl*, para acessar as portas e receber dados para à máquina com amplo suporte de desenvolvimento, possui entrada USB e placas de circuitos que podem se conectar para toda ampliação eletrônica, no caso do protótipo CNC, o Arduino UNO conecta a uma placa CNC *Shield*, específica em se comunicar com os drivers de potência dos motores de passo e toda alimentação elétrica do protótipo.

7.2.1.2 CNC Shield

No Arduino UNO pode-se conectar através de fios em direto aos motores de passo ou elemento eletrônico do protótipo, sendo que os *drivers* de potência ou botões ou até mesmo chaves de liga e desliga, no entanto, se adapta a esta ligação de forma incomplexa e sucessível a erros de comunicação (figura 23).

Figura 23 - Arduino UNO ligados aos drivers de potência



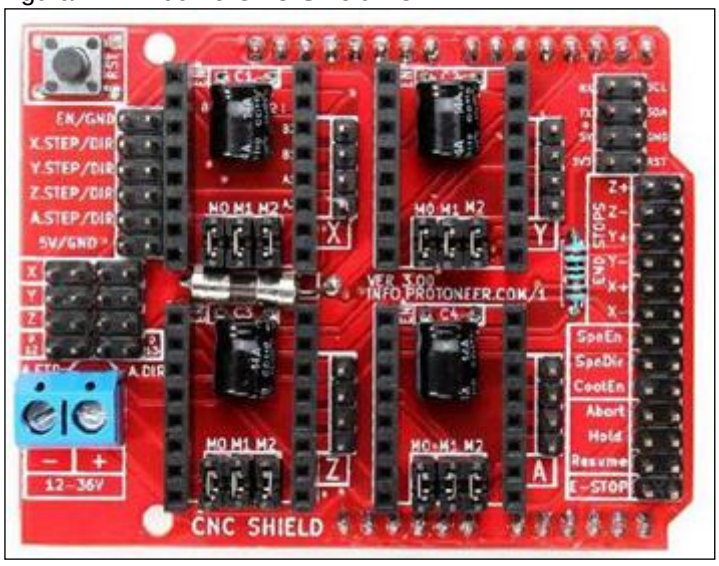
Fonte: Do autor.

Por motivo de organizar os elementos eletrônicos e elétricos conectados, utiliza-se uma placa pronta junto ao Arduino UNO que se chama CNC S*hield*, a fim de evitar falhas na comunicação de dados.

A CNC *Shield* é uma placa de circuitos de fácil conexão com a placa Arduino UNO ou com outra placa CNC S*hield*, quando conecta ao Arduino o mesmo expande sua capacidade em integrar *displays*, cartões de memória ou módulos como por exemplo o *bluetooth*.

Neste protótipo se utiliza o Arduino CNC *Shield V3* (figura 24), para conectar aos *drivers* de potência, Arduino UNO, motores de passos e fonte de energia.

Figura 24 - Arduino CNC Shield V3



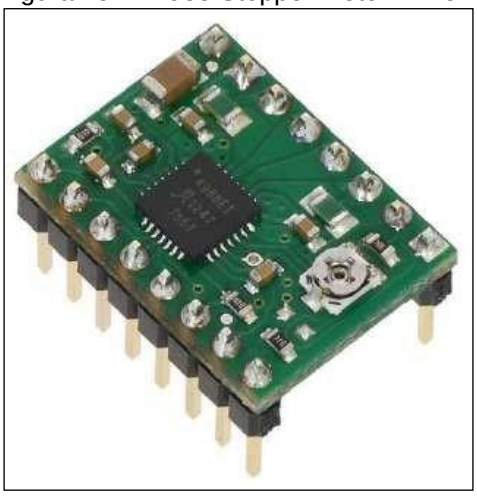
Fonte: Do autor.

7.2.1.3 Drivers de potência

Os *drivers* de potência têm a função em transformar sinais elétricos de baixa potência por sinais de alta potência, dentro dos limites de tensão e corrente, sendo necessário o uso da alimentação no dispositivo maior que a nominal do sistema que se controla.

No protótipo o *driver* de potência é responsável por ligar os motores de passo, o *driver* ideal que se utiliza é o *A4988 Stepper Motor Driver* (figura 25).

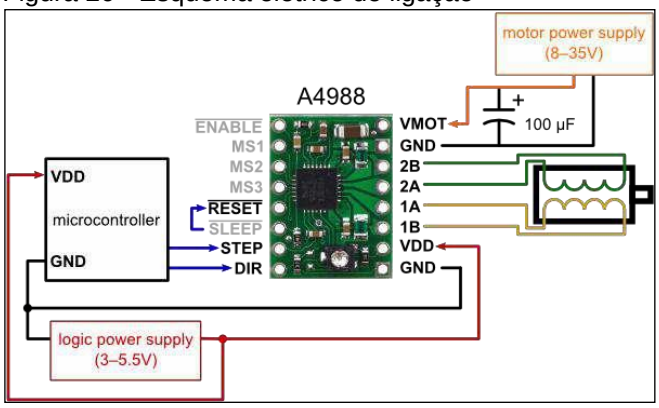
Figura 25 - A4988 Stepper Motor Driver



Fonte: Do autor.

O *driver* controla os motores de passo e pode trabalhar entre tensões de 8 e 35 volts, sendo assim, entrega até 1,5 Ampères por bobina, na figura 26 representa o esquema elétrico de ligação.

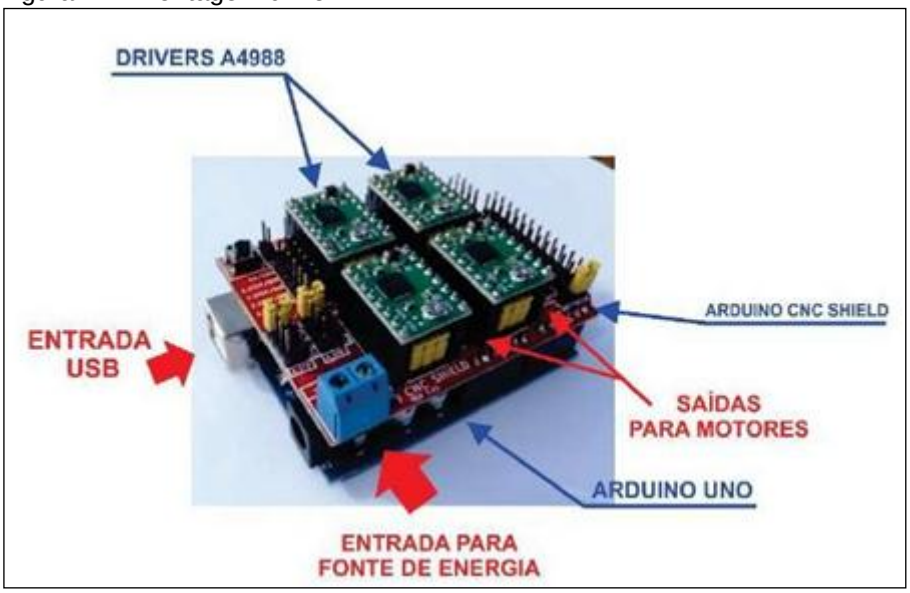
Figura 26 - Esquema elétrico de ligação



Fonte: Do autor.

O esquema eletrônico é simples e prático entre o *driver* e CNC *Shield* pois já tem posição de encaixe própria para o *driver A4988*,de acordo com a figura 27, obtendo assim a placa principal conjunta do protótipo.

Figura 27 - Montagem driver



Fonte: Do autor.

7.3 FONTE DE ALIMENTAÇÃO

A fonte de alimentação fornece toda a demanda de energia necessária para o protótipo, deve calcular a demanda energética que se utiliza, em todos os elementos elétricos e eletrônicos, para evitar insuficiência energética ou queima de componentes com tensão acima de suas limitações.

Para ter uma fonte que forneça a demanda ideal de energia para o protótipo, precisa respeitar o dimensionamento energético de cada componente conforme a tabela 1.

Tabela 1 – Demanda energética

Componente	**Quantidade**	**Tensão Operante**	**Consumo**	**Consumo Total**
Arduino UNO	1	7 a 12 V	800mA	800mA
Arduino CNC *Shield*	1	12 a 36V	--	--
Driver de potência	3	8 a 35 V	50 mA	150mA
Motor de passo	3	5 a 36 V	1.5 A	4.5 A
Demanda	**12 V**		**máximo 9A**	

Fonte: Do autor.

Observa-se na tabela acima, na placa Arduino UNO recomenda-se uma tensão para operar entre 7 a 12 volts e os *drivers* de potência deve operar na faixa de tensão entre 8 a 35 volts, optou-se por utilizar uma fonte de energia que forneça uma tensão nominal na saída com12 volts.

Outro fator importante é a corrente mínima necessária para funcionar o protótipo com cada motor de passo onde consome 1,5 Ampères com corrente mínima somadas a 6 Ampères, neste caso todos os motores devem acionar simultaneamente um acréscimo de 1 Ampère para funcionar os demais componentes eletrônicos.

Assim são 9 Ampères em corrente mínima para o protótipo funcionar e se adiciona uma margem de segurança de 30% na corrente de 1,5 Ampères.

Com estas informações pode-se determinar uma fonte de alimentação de 12 Volts.

A fonte de alimentação selecionada para o protótipo foi a fonte padrão *ATX*

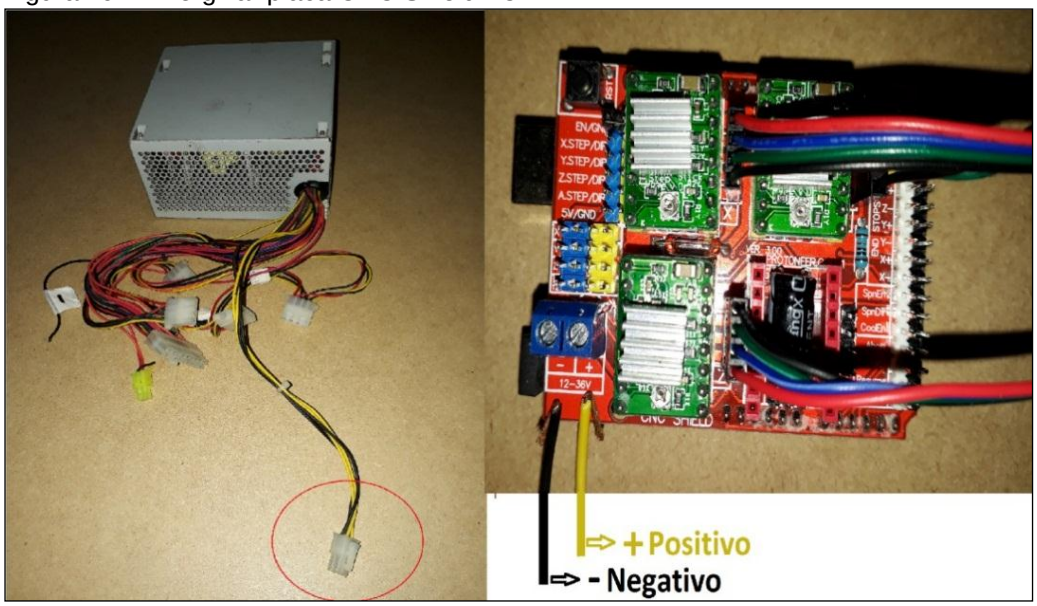
utiliza-se em microcomputadores, com 264 Watts de potência e saídas de tensão de +3.3V, +5V, +12V e -12V, na saída de +12V fornece correntes de até 14 Ampères (figura 28).

Figura 28 – Fonte ATX

Fonte: Hardline (2018).

Usa-se o conector de alimentação da fonte *ATX* para energizar a placa CNC *Shield V3*, sabe-se que as cores dos fios indicam suas polaridades, o fio de cor amarela indica o pólo positivo e o fio na cor preta indica o pólo negativo, deste modo, conecta-se os fios na alimentação da placa CNC *Shield V3* (figura 29).

Figura 29 – Energizar placa CNC Shield V3



Fonte: Do autor.

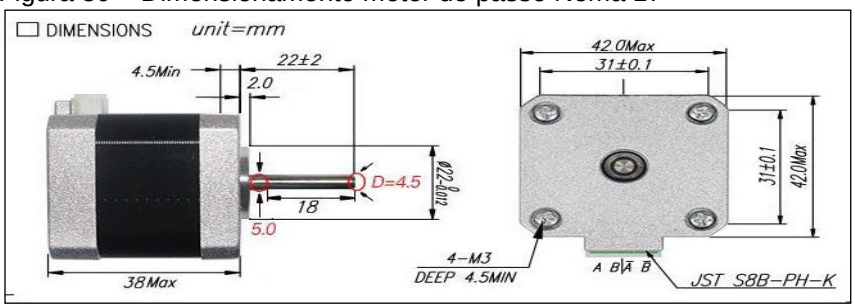
Como se trata de alimentação em baixa corrente, não é necessário neste protótipo de refrigeração com ventilação na placa CNC *Shield V3*.

7.4 MOTORES DE PASSO

Conhecido como atuadores os motores de passos têm o princípio de converter um dado tipo de energia em sentido de giro, no caso deste protótipo, o sentido de giro se torna em sentido de movimento linear pelos eixos X, Y ou Z.

Os motores de passo utilizado neste protótipo é um *Nema 17* consome 1,5 Ampères por atuador e dimensão ideal para o projeto (figura 30).

Figura 30 – Dimensionamento motor de passo Nema 17



Fonte: Datasheet (2018).

Os *drivers* de potência devem controlar todo fluxo de energia em cada fase do motor de passo deste projeto, a capacidade energética se calcula conforme a corrente e tensão nominal do atuador (tabela 2), os motores de passo que se utiliza têm dimensionamento padrão *Nema 17*, com 39 N.cm (Newton por centímetro) de torque, passo de 1.8 graus e corrente máxima de 1,5 Ampères por fase.

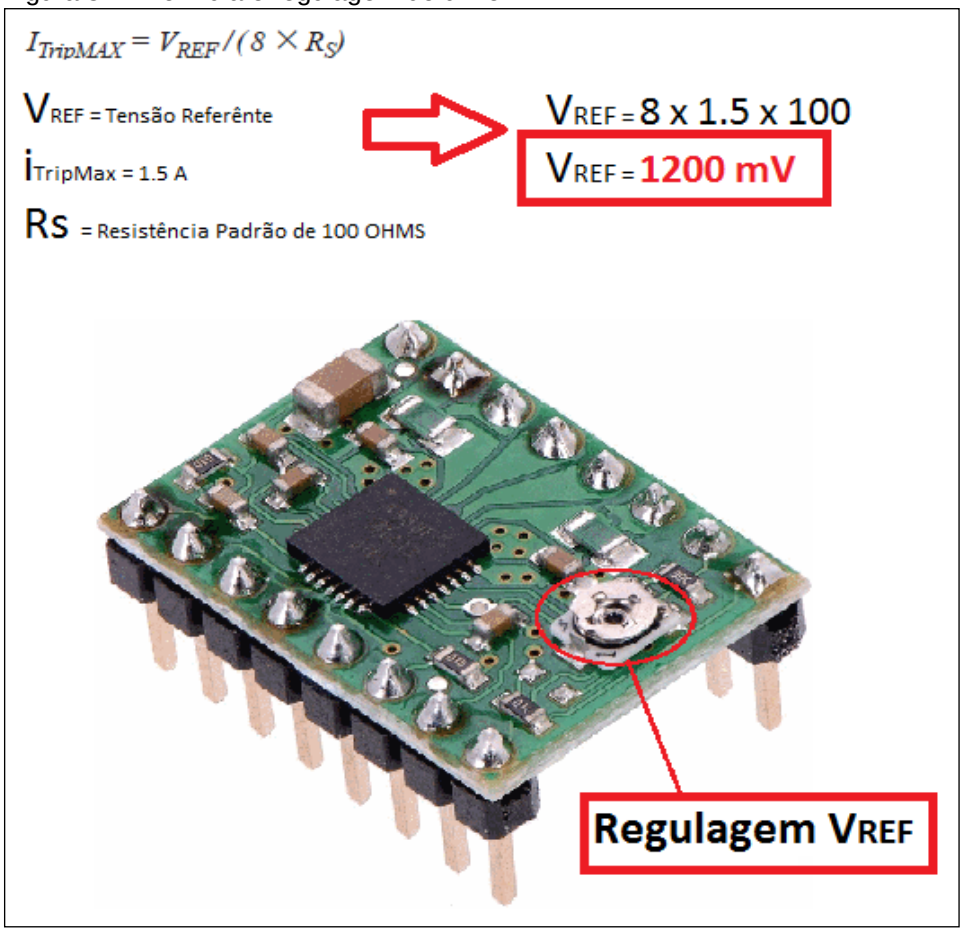
Tabela 2 – Especificações Técnicas *Nema 17*

Especificação	**Fase**
Ângulo	1.8
Voltagem	2.4 V
Corrente	1.5 A/Fase
Resistência	1.6 ohms / Fase
Indução	3.0 mH/Fase
Torque	39 N.cm

Fonte: Datasheet (2018).

Com os dados de referência elétricos dos fabricantes do motor de passo Nema 17 e do driver A4988, através da fórmula do fabricante do driver (figura 31), regula-se a tensão de corrente contínua nominal ideal de trabalho resultante a 1200 mV, com multímetro.

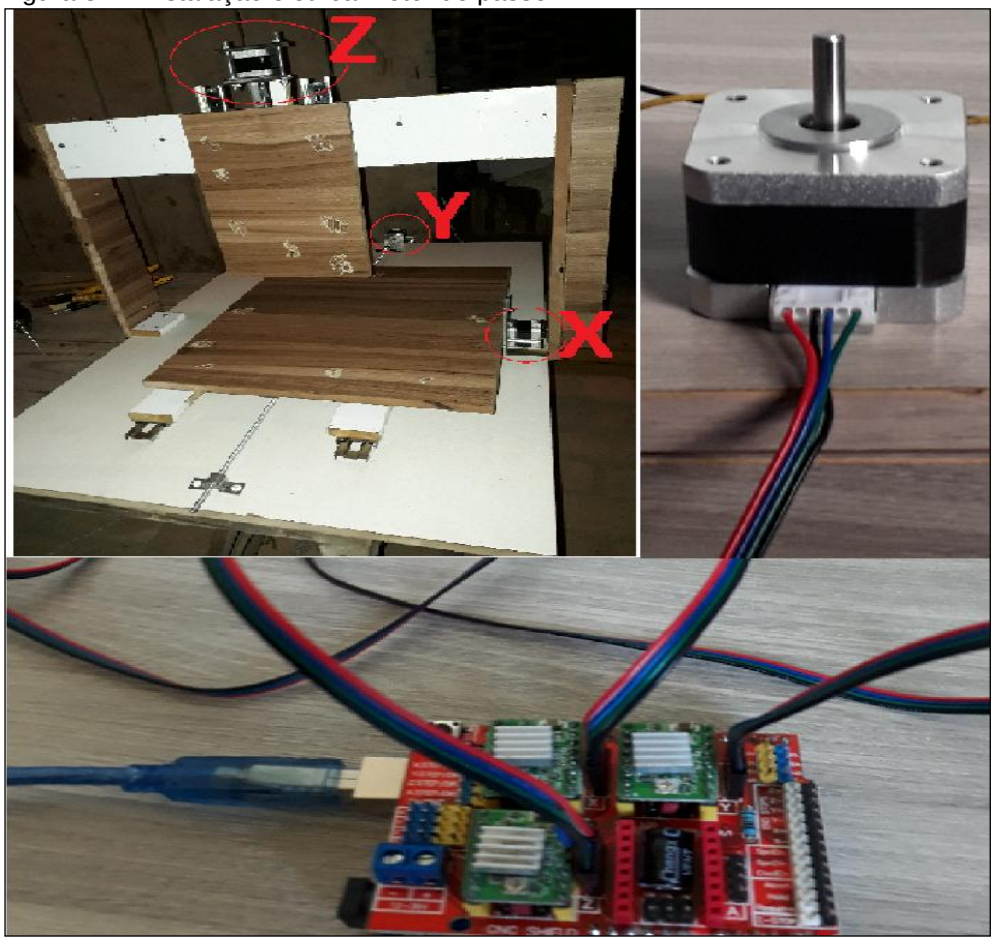
Figura 31 – Fórmula e regulagem do driver



Fonte : Allegromicro (2018).

No protótipo o motor de passo deve-se instalar na base da máquina e encaixar no extremo da barra roscada onde consta o acoplamento, fixo com suporte na carcaça do motor para não mover no momento de operação, a alimentação elétrica parte da CNC *Shield V3* com cabo de 4 vias conforme a figura 32.

Figura 32 – Instalação elétrica motor de passo



Fonte: Do autor.

7.5 EIXO ÁRVORE

A ferramenta de corte tem a função de cortar e transformar matéria prima em peça, em uma R outer utiliza-se um motor de alta rotação em seu eixo uma haste de fixação ou pinça de pega, que segura a ferramenta cortante ou fresa.

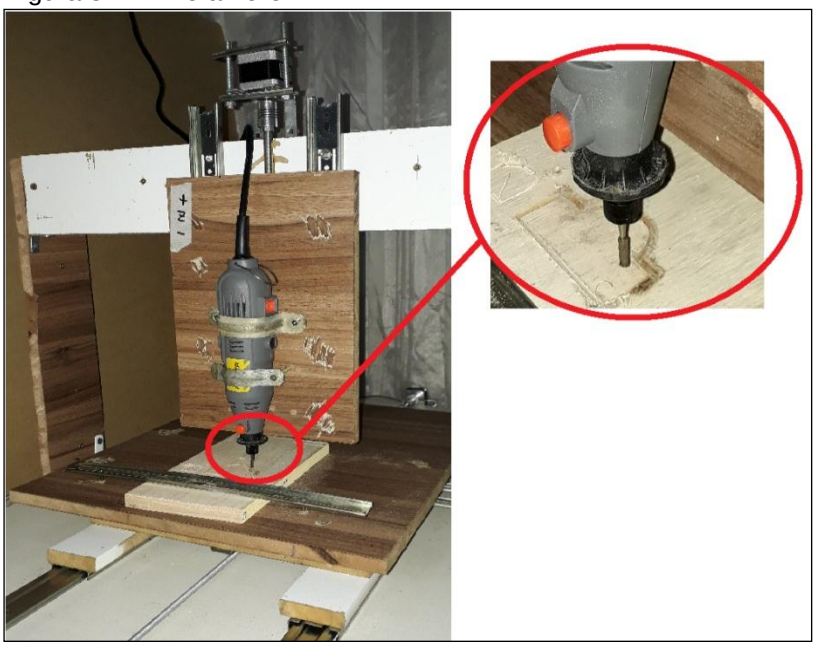
Neste protótipo máquina CNC foi adaptado uma micro retífica com velocidade variável de 8.000 a 33.000 RPM de 135 Watts de potência e 220 Volts com pinça para pega de fresa de diâmetro 3.2 mm com comprimento de 9.5 mm de 5 arestas cortantes, por se tratar de usinagem em materiais leves esta retífica atende à demanda e traz resultados satisfatórios (figura 33).

Figura 33 - Micro retifica e fresa

Fonte: Do autor.

A micro retífica é fixada na base do eixo Z, onde se tem a função de eixo árvore conforme figura 34.

Figura 34 – Eixo árvore



Fonte: Do autor.

8 SOFTWARES E LINGUAGEM CNC

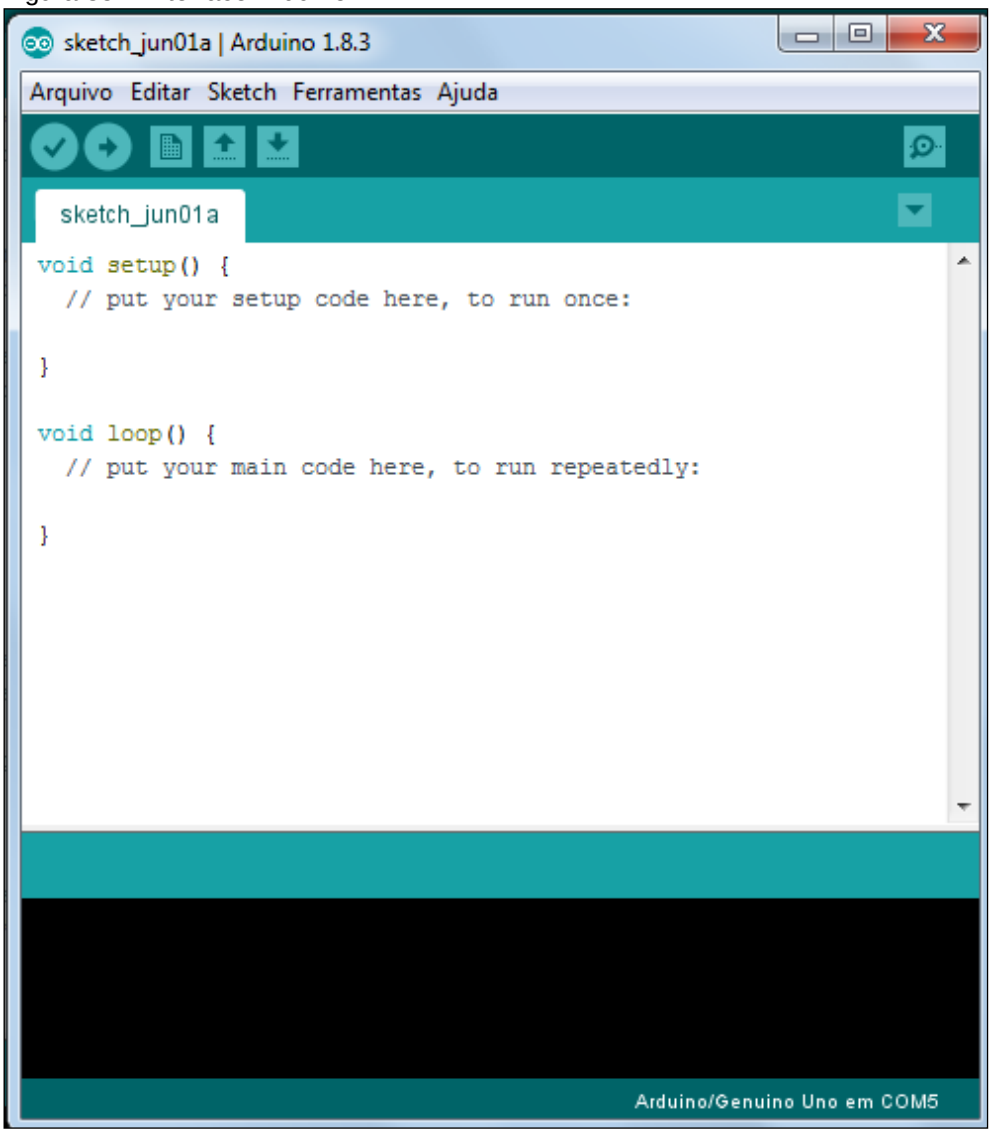
O protótipo CNC se comunica via cabo ligado ao Arduino Uno e computador, e para realizar os parâmetros de linguagem onde o Hardware irá obedecer e deslocar, usa-se o software Arduino para programar a CNC Shield V3 e para desenhar peça que será feita com função CAD usa-se o software SketchUp 2016 com biblioteca de ferramentas CAM adicionadas para gerar a linguagem CNC, conhecida no mundo de máquinas *Router* como G-Code e seu Software de leitura da linguagem CNC G-Code é o Universal Gcode Sender, onde têm o objetivo principal em dar coordenadas em medidas precisas para os motores de passo de cada eixo, formando assim a peça mensurada em desenho intangível em uma peça real ou tangível.

Todas as plataformas utilizadas neste protótipo são livres ou open source, encontra-se em sites de busca ou sites de máquinas caseiras com Arduino UNO, exige apenas entendimento e conhecimento no mercado de máquinas operatrizes CNC, softwares CAD e CAM e programação eletrônica.

8.1 SOFTWARE ARDUINO

Como usa-se o hardware Arduino UNO, o software Arduino é que opera e programa sobre esta placa, um software de aplicação em multiplataforma escrita em Java com recursos de sintaxe, parênteses e identificação automática, compila e carrega os programas desenvolvidos para a placa de forma fácil, algumas bibliotecas dão a capacidade de programar em C ou C++, ambiente requer duas seções para funcionamento da programação na placa Arduino UNO, *VOID SETUP*, são as configurações que inicia os componentes eletrônicos que irá operar no programa e *VOID LOOP*, faz repetir o comando programado no bloco até que o usuário finalize a seção ou desliga (figura 35).

Figura 35 – Interface Arduino

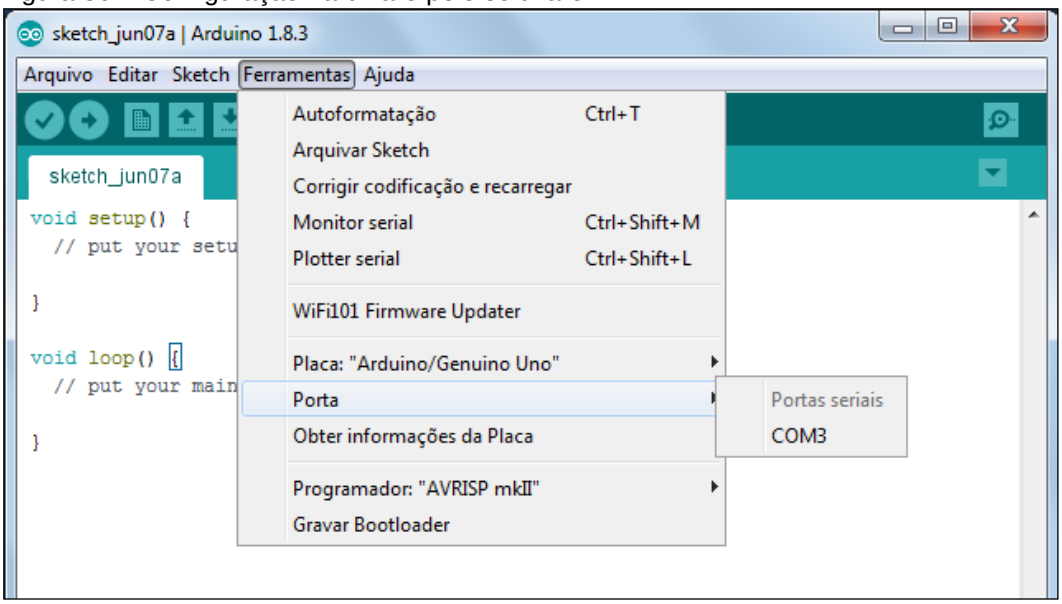


Fonte: IDE Arduino (2018).

8.1.1 Configuração software Arduino

Para enviar programas do software Arduino para a placa Arduino Uno, basta-se conectar o hardware no computador via cabo USB, identifica-se a porta COM (USB) e modelo da placa Arduino/Genuíno UNO na aba Ferramentas (figura 36).

Figura 36 – Configuração hardware pelo software



Fonte: IDE Arduino (2018).

8.2 FIRMWARE GRBL

O firmware *GRBL* é um código aberto desenvolvido em C, onde requer um *hardware* para operar máquinas CNC ou impressora 3D já neste protótipo usa-se Arduino UNO com a placa CNC Shield V3 para se movimentar os eixos da CNC, na figura 37 apresenta os desenvolvedores que apoiam o projeto *Grbl*.

Figura 37 - Desenvolvedores que fornecem suporte ao GRBL

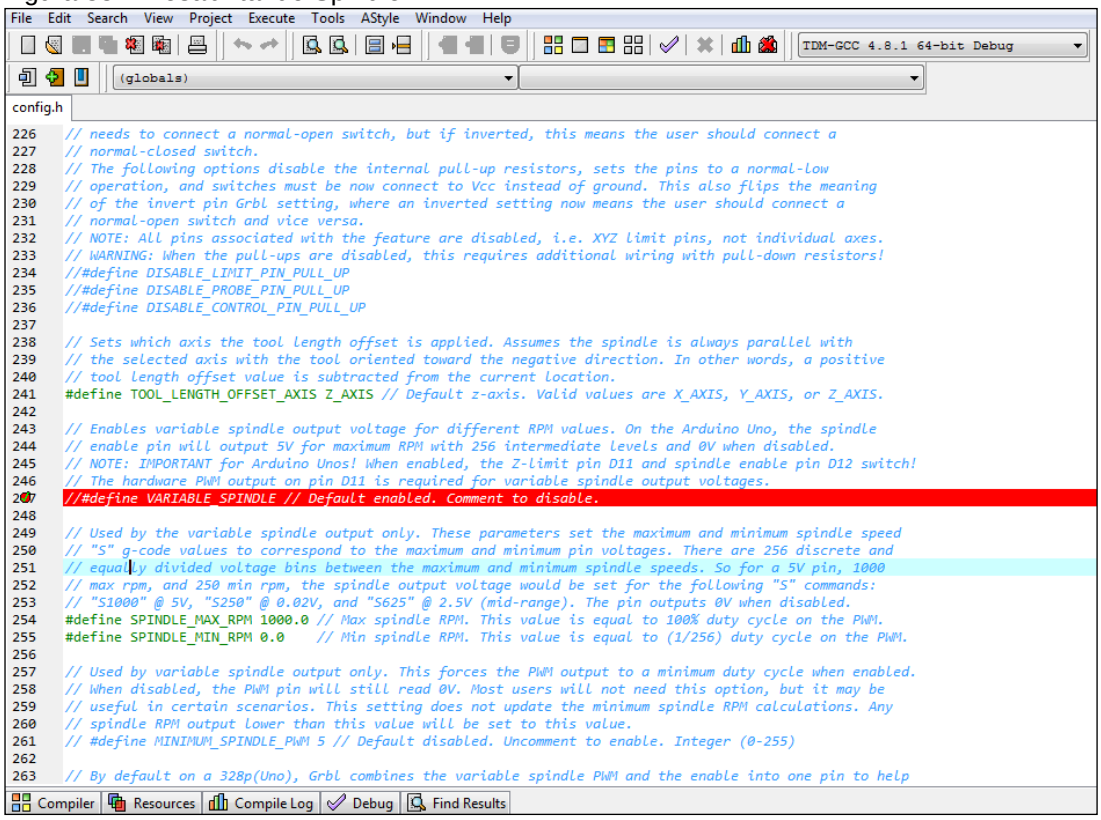
Fonte: GITHUB (2018).

8.2.1 Configuração firmware GRBL

É necessário informar ao programa uma série de parâmetros físicos e de configuração da máquina CNC para que o controle seja feito corretamente, a estratégia aplicada de maneira geral em *softwares* comerciais é a utilização de arquivos de configuração, que não obrigam o usuário a informar suas preferências a cada nova execução.

Para configurar abrir o arquivo *config* em bloco de notas, selecionar a linha *247-#define VARIABLE_SPINDLE // Default enabled. Comment to disable*, esta linha do programa define a variação de velocidade do eixo árvore, este eixo não foi instalado no protótipo, para isto precisa-se desabilitar esta função, comenta-se a linha inteira ou digitar duas vezes o caractere '// '(figura 38).

Figura 38 – Desabilitando Spindle

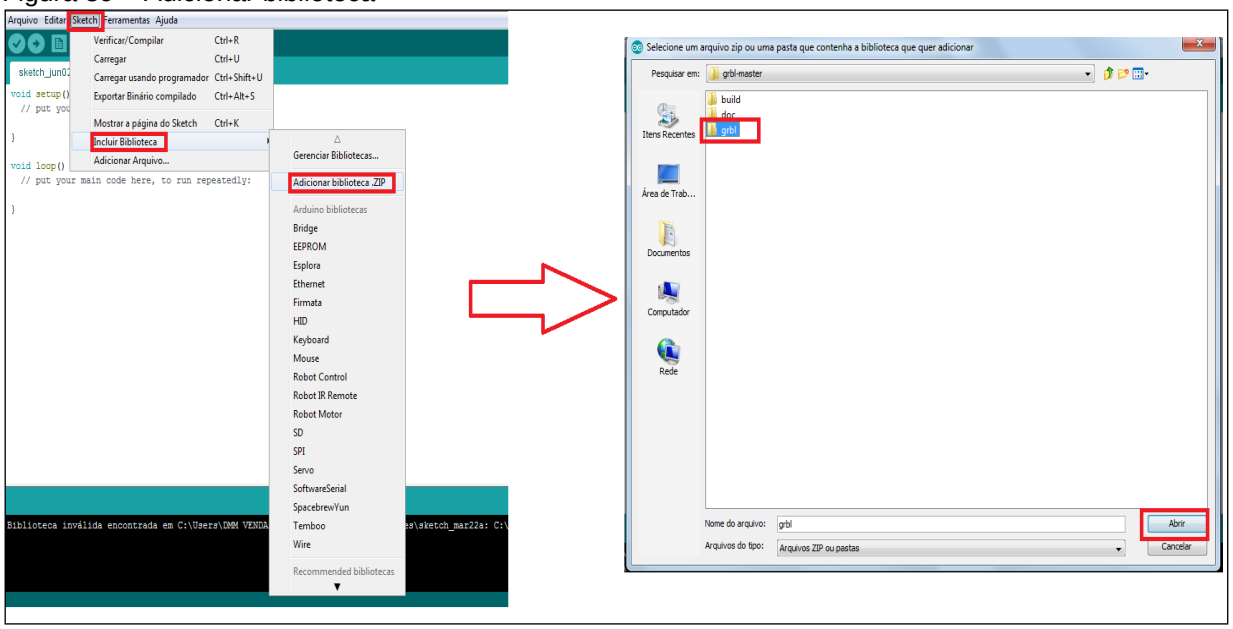


Fonte: Do autor.

A leitura de informações do arquivo *config.txt* é feita sempre que o programa se inicia, há opções para impressão dos valores lidos na tela e para realização de nova leitura a qualquer momento durante a execução, assim salvar o

arquivo *config* alterado na linha do programa, abrir o software Arduino, na aba entrar em *Sketch*, *Incluir Biblioteca* e *Adicionar Biblioteca.ZIP* (figura 39), encontrar a pasta *grbl* no diretório e abrir, este processo irá instalar a biblioteca *GRBL*.

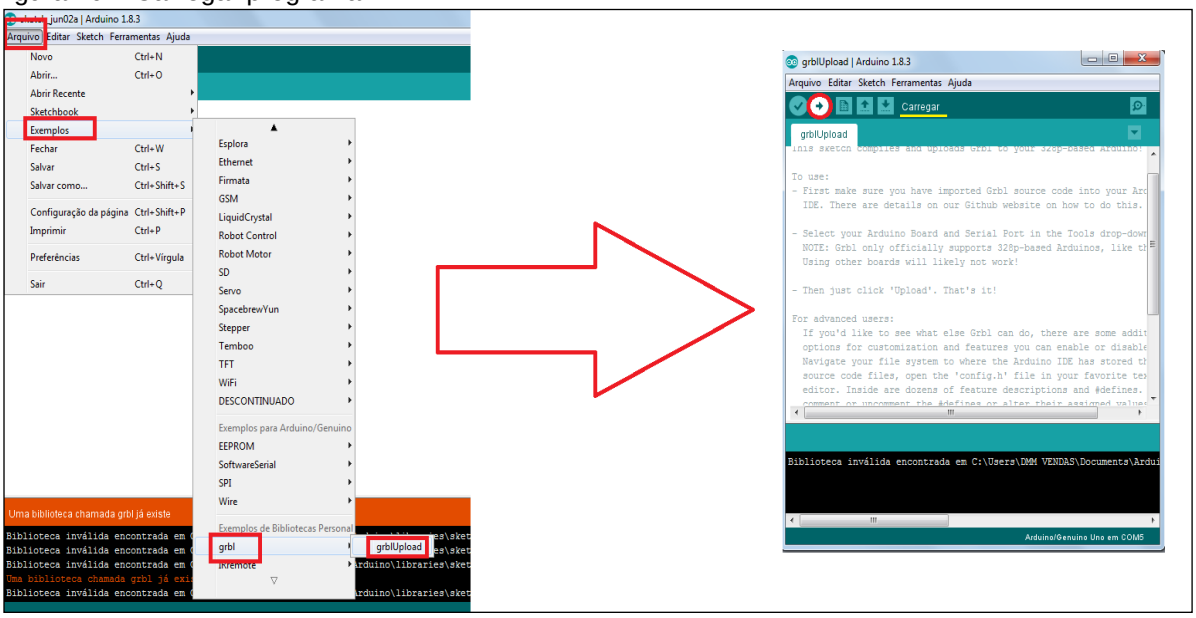
Figura 39 – Adicionar biblioteca



Fonte: IDE Arduino (2018).

Na aba *Arquivo* acessar opção *Exemplos* e selecionar a opção *grblUpload*, irá abrir uma nova janela com o programa da biblioteca *grbl*, ir em C*arregar* para enviar o programa para a placa Arduino UNO, conforme figura 40.

Figura 40 – Carregar programa



Fonte: IDE Arduino (2018).

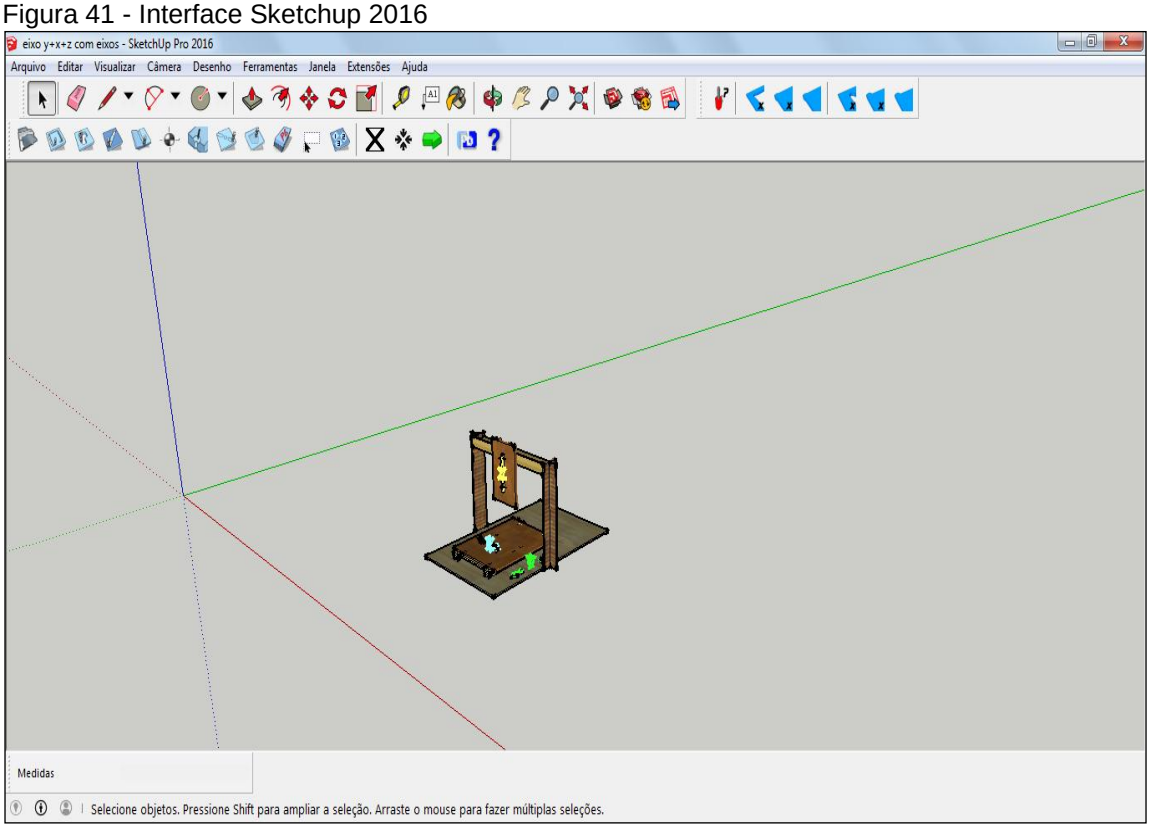
Com o *firmware GRBL* instalado na placa Arduino UNO a máquina já pode receber comandos G-CODE a fim de controlar os motores de passo.

8.3 CAD E CAM

Para criar uma peça a ser usinado no protótipo utiliza-se ferramentas para dimensionar e formar em modelo de desenho técnico com suas medidas reais e tornar estas medidas dimensionadas em linguagem programada CNC, estes softwares chamam-se Computer Aided Design e Computer Aided Manufacturing ou abrevia-se CAD e CAM, modelo que se utiliza no mercado industrial principalmente nos setores de Engenharia em Processos.

O software CAD que se utiliza neste protótipo para criar as peças é o Sketchup 2016, além se tratar de um software gratuito também é fácil de operar, ferramentas clássicas para desenho computadorizado como o lápis digital, linhas, largura, espessura e figuras rápidas como círculo e quadrado em formatos de objetos 2D e 3D e também instalar bibliotecas para implementar funções extras nos projetos, figura 41 mostra a interface gráfica do software.

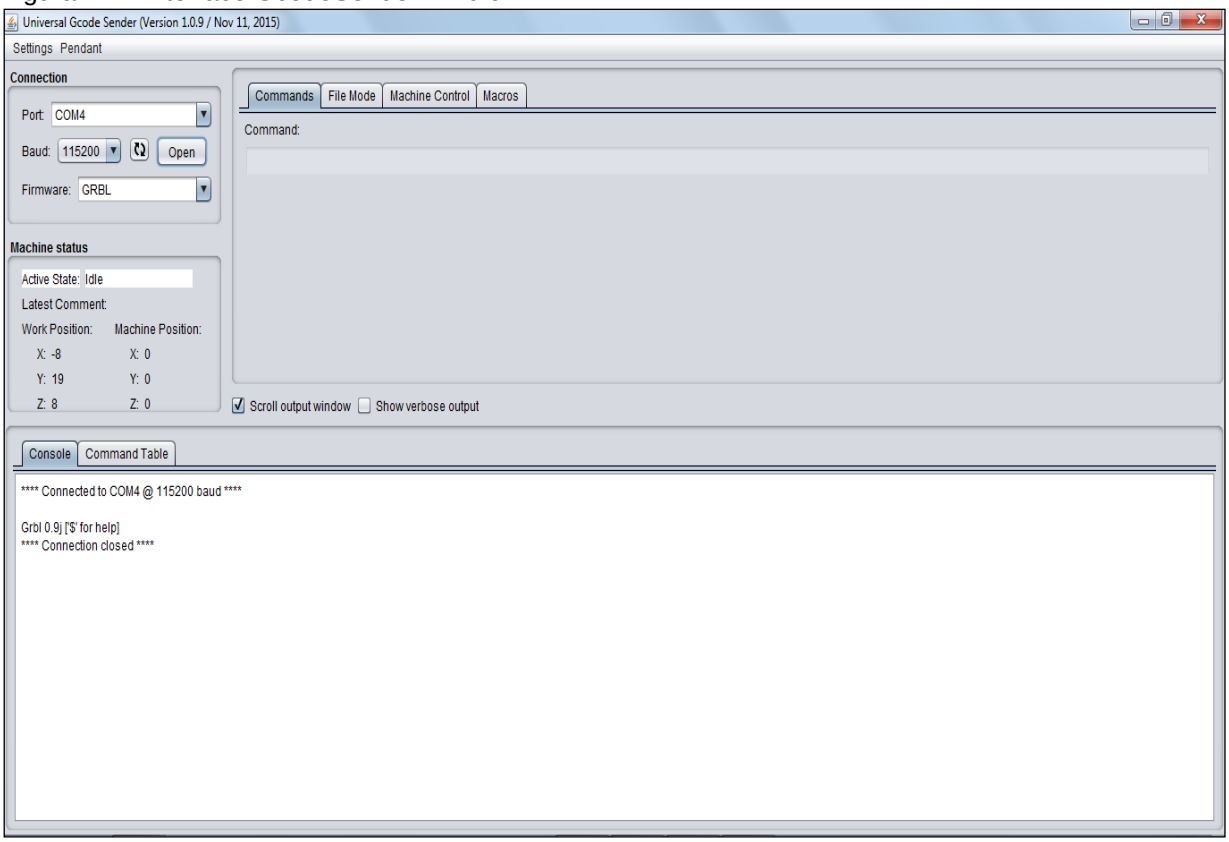
Figura 41 - Interface Sketchup 2016



Fonte: IDE Sketchup (2018).

Para se movimentar e manufaturar peça no protótipo máquina CNC usa-se o software CAM gratuito *Universal Gcode Sender V 1.0.9*, cria-se todo o processo de produção dentro do que se configurou entre software e hardware, e com base na programação CNC, inicia-se o deslocamento de acordo com o que se dimensionou em CAD, pode-se simular a trajetória de usinagem operatriz antes mesmo de usinar e também operar em modo manual em eixos independentes (figura 42) .

Figura 42 – Interface GcodeSender V1.0.9

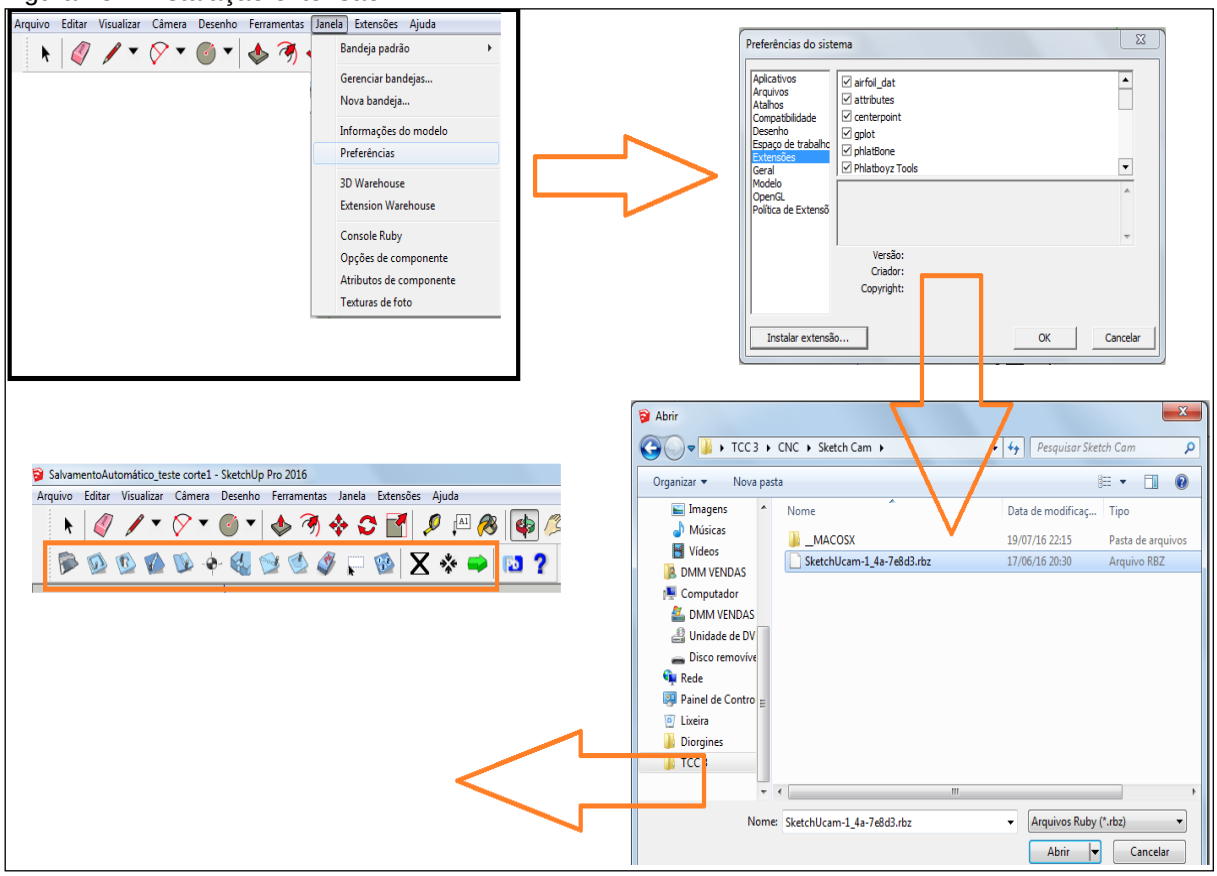


Fonte: IDE Gcode Sender (2018).

8.3.1 Configuração Sketchup

O software Scketchup para funcionar no protótipo deve-se instalar a extensão adicional SketchUCAM que se encontra no site da *OPENBUILDS*, para instalar tem que acessar a opção *janela*, *instalar extensão* em *preferências do sistema*, encontrar o diretório do arquivo e abrir *SketchUCAM*, conforme figura 43, neste modo se adiciona novas ferramentas na área de trabalho do software.

Figura 43 – Instalação extensão



Fonte: Do autor.

Na extensão SketchUCAM se ajusta os parâmetros de manufatura do desenho com a máquina, sendo que o desenho peça tem-se que se usinar dentro do limite espacial de área do protótipo, abrir a ferramenta *phlatboyz parameters* e configurar, conforme a figura 44, todas as funções usais:

a) *Feed Rate*, velocidade de corte entre X e Y em milímetros por segundo;

b) *Plunge Rate*, velocidade do eixo Z com a mesa em milímetros por segundo;

c) *Material Thickness*, espessura do material bruto em milímetros;

d) *In/OutsideOvercut*, porcentagem de corte configurar mediante folgas no eixo Z;

e) *Bit Diameter*, diâmetro da fresa em milímetros;

f) *TabWidth*, espessura do material que fica entre a peça e o material bruto na largura em milímetros;

g) *TabDepth*, espessura do material que fica entre a peça e o material bruto na espessura em por cento;

h) *Safe Travel Z*, subir eixo em relação ao ponto zero máquina em milímetros;

i) *Safe Lenght X*, dimensão de curso em X em 150 milímetros;

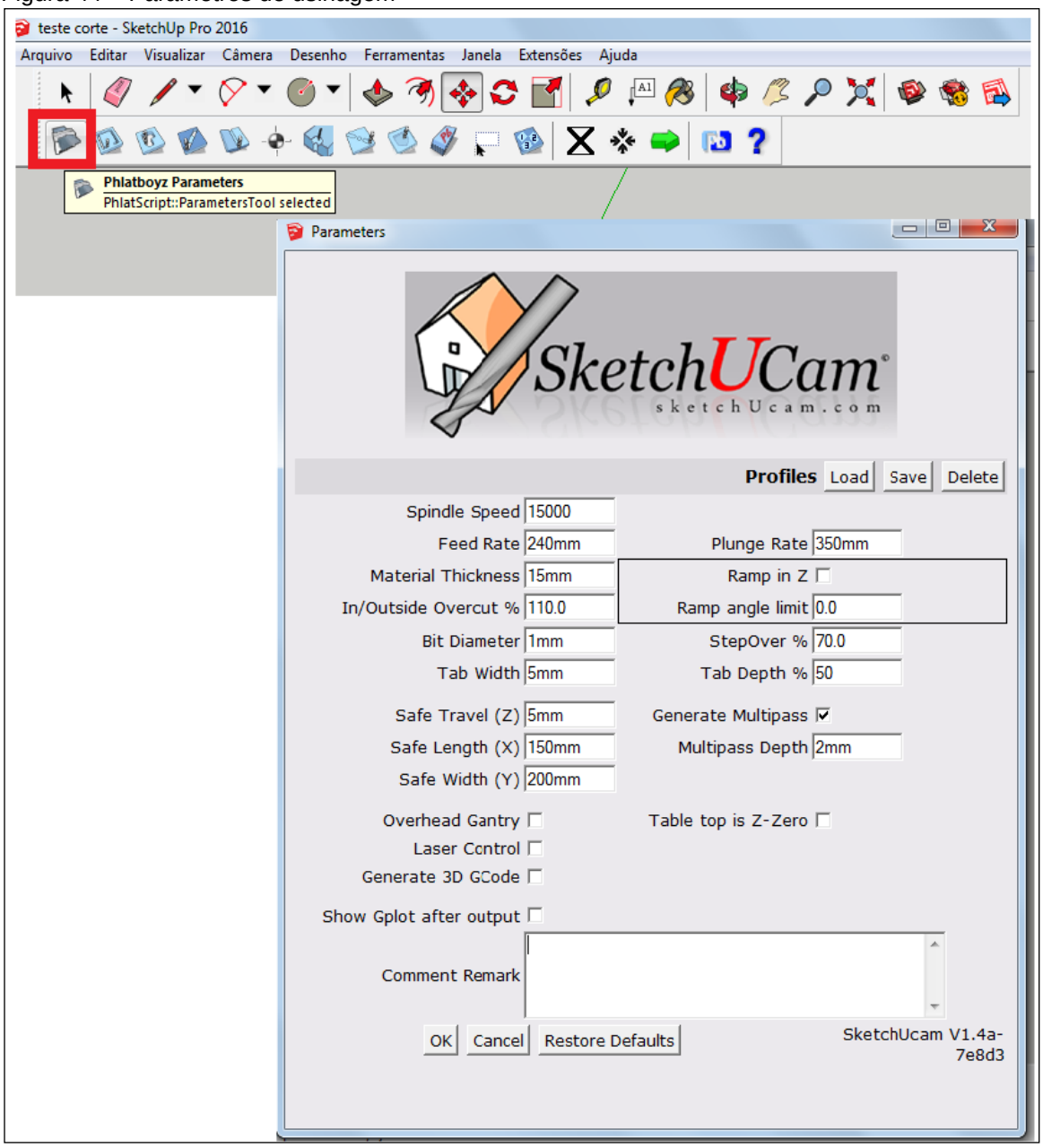
j) *Safe Width Y*, dimensão de curso em Y em milímetros;

k) *Step Over*, material que a fresa não corta por passada manter o padrão em porcento;

l) *GenerateMultipass*, gerar passos de usinagem;

m)*MultipassDepth*, espessura para usinar a cada passada em milímetros.

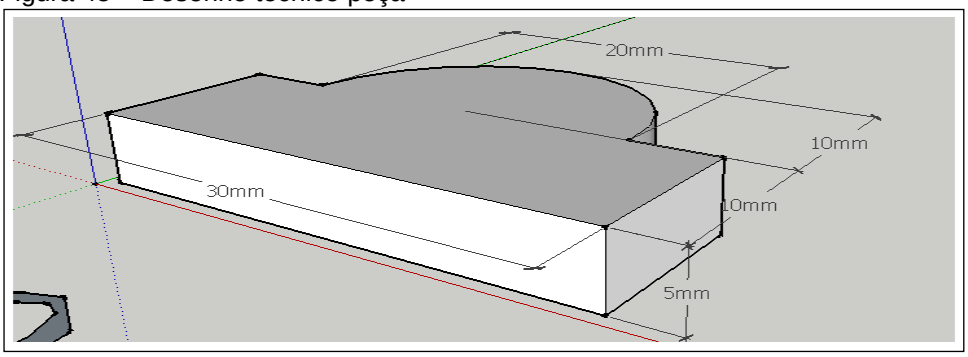
Figura 44 – Parâmetros de usinagem



Fonte: Do autor.

Com o Sketchup configurado, projeta-se a peça com suas dimensões através das ferramentas de desenho CAD dentro da área útil de fabricação do protótipo (figura 45).

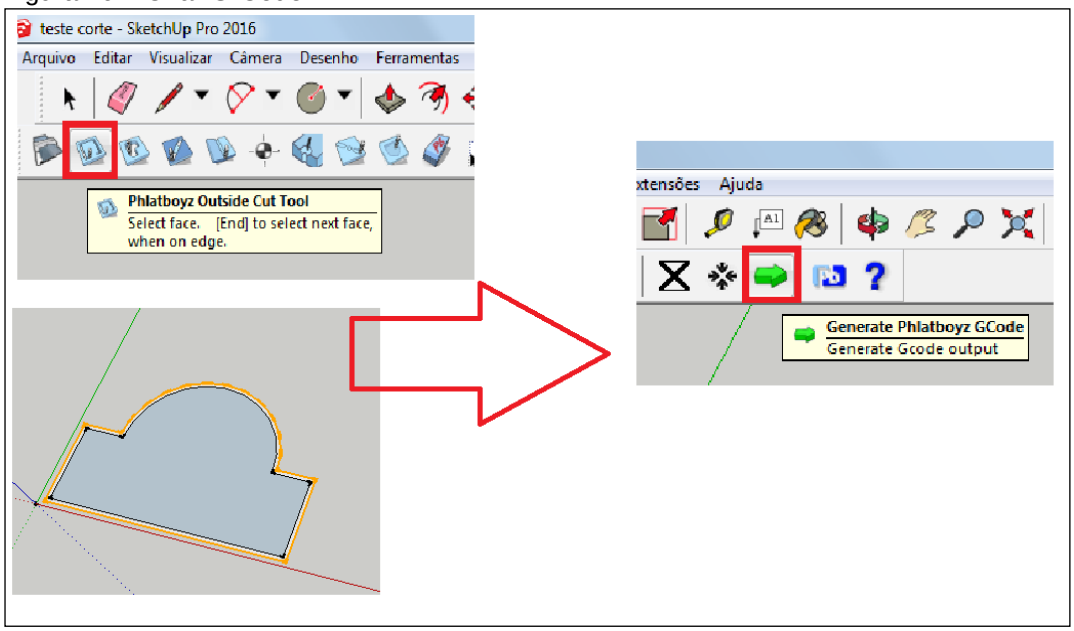
Figura 45 – Desenho técnico peça



Fonte: Do autor.

As peças são dimensionadas em milímetros (mm) e com o desenho da peça pronta deve-se criar o G-Code das linhas de medição de cada aresta do desenho, seleciona-se a ferramenta *Phlatboyz Outside Cut Tool*, define-se as linhas de usinagem com a ferramenta de seleção e na opção *Generate Phlatboyz GCode* gera-se a coordenada de linguagem CNC ou G-Code para leitura e operar a máquina, conforme figura 46.

Figura 46 – Criar G-Code

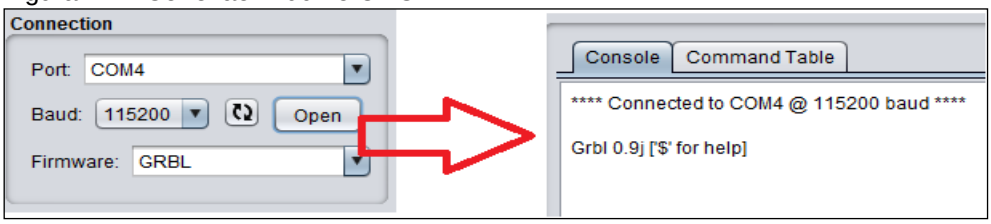


Fonte: Do autor.

8.3.2 Configuração Gcode Sender

Para acessar ou configurar funções no software *Universal GcondeSender* tem-se que conectar com o hardware *Arduino UNO* via cabo *USB* e placa *CNC SHIELD V3* ligado na *Fonte ATX*, no software na opção *CONNECTION*,selecionar a porta *COM* (*USB*), identificar em *BAUD* a velocidade que deve se comunicar com o firmware *GRBL*, conforme figura 47, clicar no botão *OPEN* na aba *CONSOLE* onde deve-se aparecer a mensagem *Grbl 0.9j ['$' for help]* , comando conectado com o *Arduino UNO*.

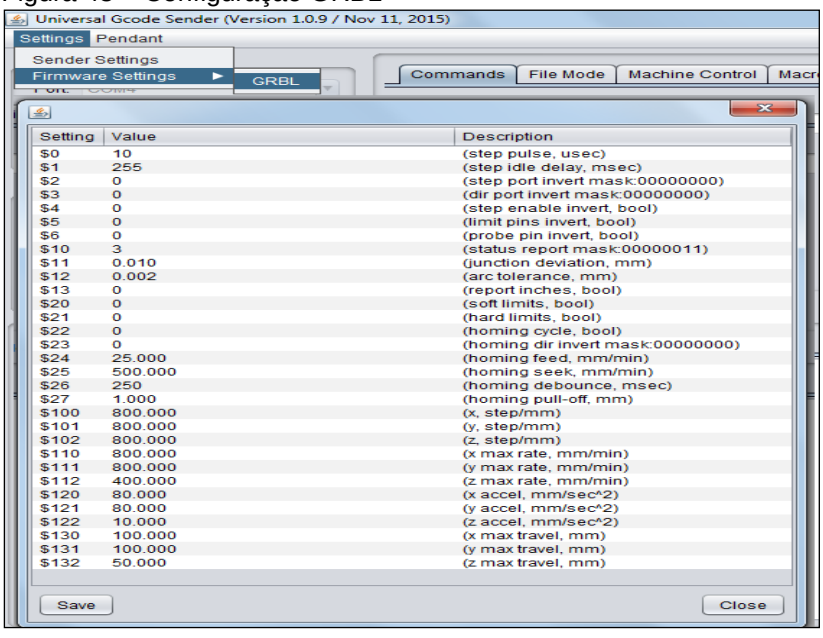
Figura 47 – Conexão Arduino UNO



Fonte: Do autor.

Com o hardware conectado pode-se configurar o software Universal Gcode Sender com os parâmetros característicos do protótipo para isto deve-se ir na aba *SETTINGS* acessar *FIRMWARE SETTINGS* e abrir a opção *GRBL* (figura 48).

Figura 48 – Configuração GRBL

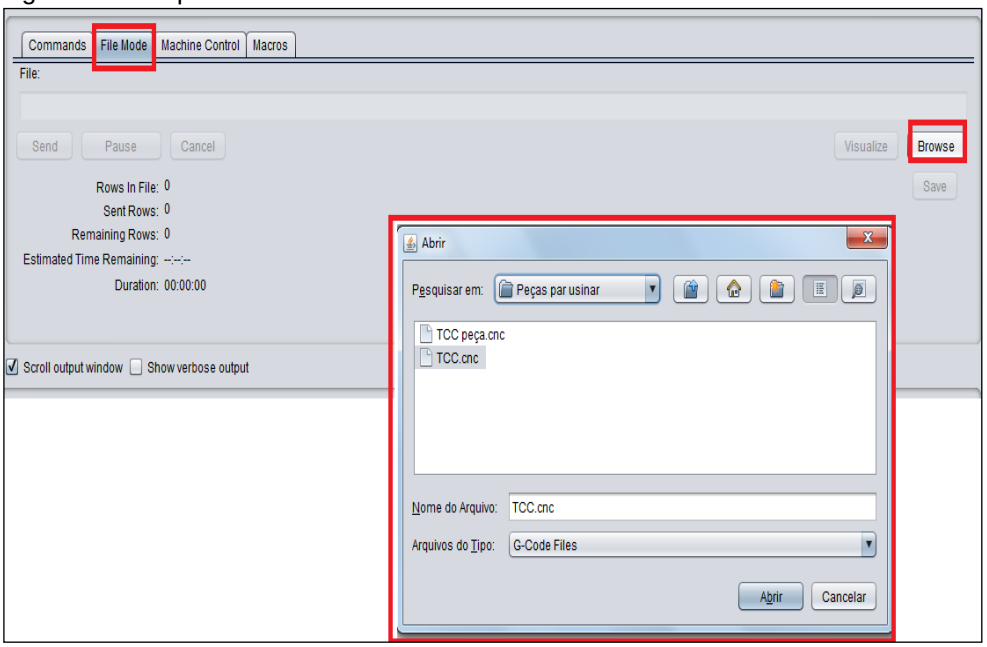


Fonte: Do autor.

Na janela de configurações *GRBL*, altera-se os valores em *VALUE*, seguidos de sua ordem em *SETTING* conforme cada opção em *DESCRIPTION*, para realizar o dimensionamento do melhor ajuste no protótipo foi utilizado como base a apostila de configuração que está em anexo 1.

Na aba de operação entrar em *FILE MODE* e acessar o *BROWSE*, encontrar o diretório do arquivo com extensão em *G-Code* salvo pelo *Scketchup*, conforme figura 49.

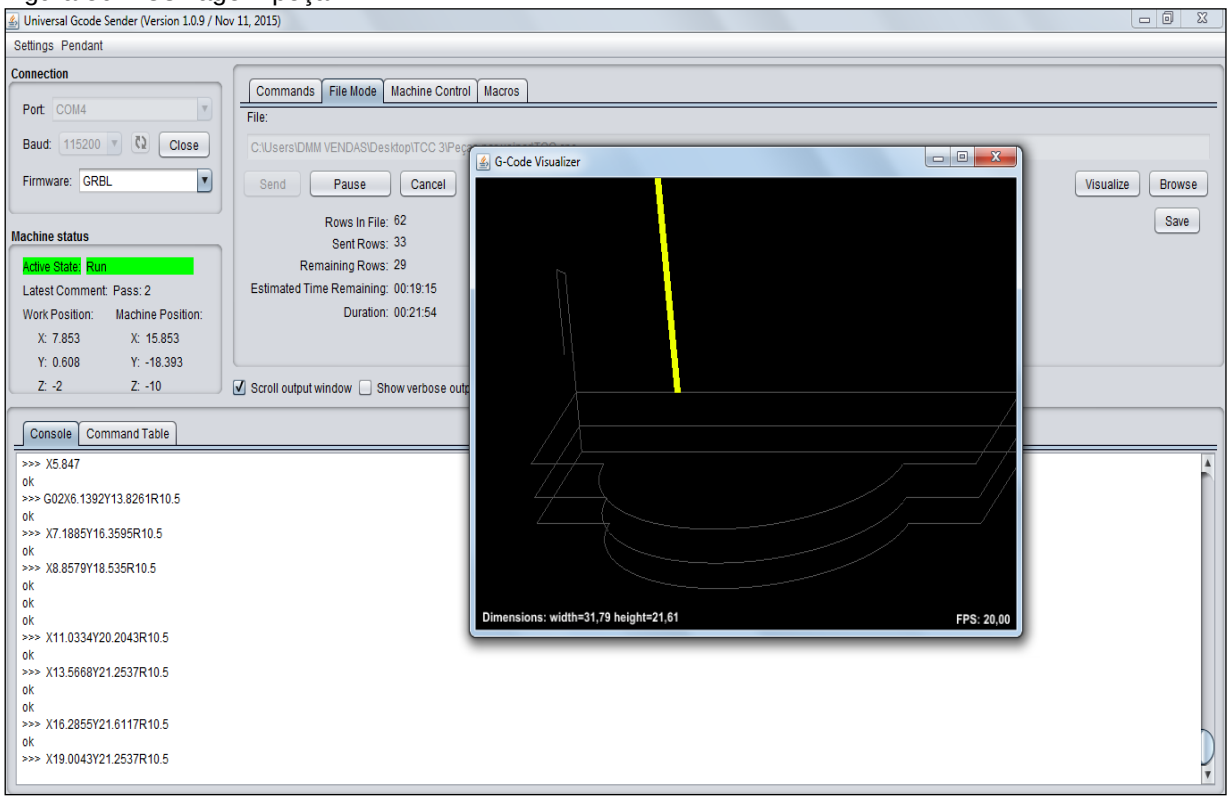
Figura 49 – Arquivo G-Code



Fonte: Do autor.

No protótipo cada eixo se posiciona em ponto zero ou ponto inicial, com o arquivo G-Code aberto no software, pode-se visualizar simulação gráfica de todo o processo de usinagem e as linhas da linguagem de programação no apêndice B, em *SEND* inicia-se o processo de conformação da peça (figura 50).

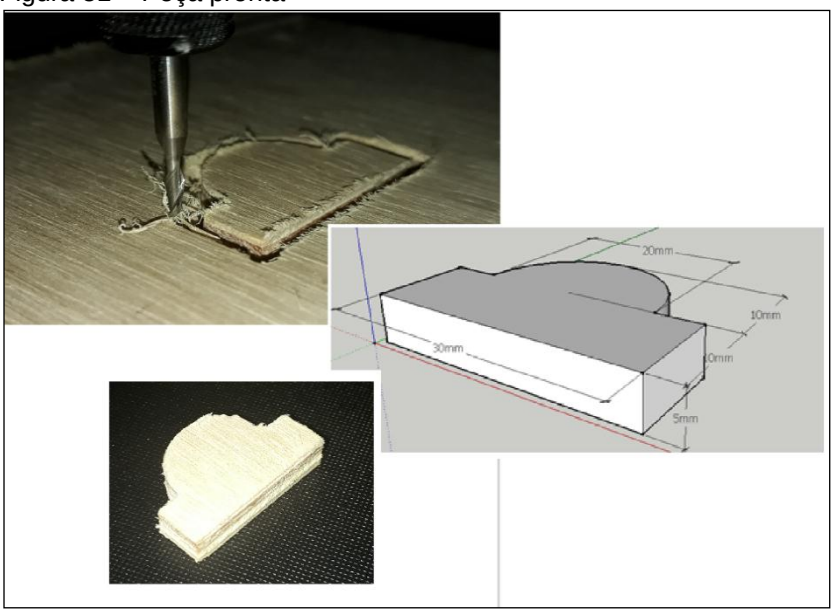
Figura 50 – Usinagem peça



Fonte: Do autor.

A máquina finaliza seu processo de usinagem e retorna automaticamente para seu ponto zero, o resultado é uma peça pronta com suas dimensões mensuradas (figura 51).

Figura 51 – Peça pronta



Fonte: Do autor.

9 PROGRAMAÇÃO CNC

Para geração de dados de programação no protótipo é gerado através da extensão Sketch UCAM no software Sketchup, onde transforma linhas de medidas do desenho em comando de programa em formato de texto obedecendo o sistema de código padrão ‘G’ ou G-Code, este programa determina as coordenadas cartesianas com diversas colunas de pares ordenados que permitem ao usuário, com o auxílio do programa Universal Gcode Sender, visualizar graficamente as trajetórias calculadas pelo programa e sinais de controle enviados, via porta paralela, para a placa de acionamento da máquina CNC em utilização.

A programação CNC compreende a preparação dos dados para usinagem da peça pela máquina, através do arquivo do programa da peça em G-Code com informações contidas no programa que são referentes às dimensões e qualidades da peça, parâmetros de usinagem, funções de máquina, dados de ferramentas e informações tecnológicas, que permitam que a máquina produza a peça de forma automática.

9.1 FUNÇÕES GCODE

A estrutura de um programa CNC baseia-se em funções composta de:

a) cabeçalho de programa: ligação dos eixos e configuração da área de usinagem, espessura do material e medidas da ferramenta de corte;

b) comentários: informação não executáveis, detalhes específicos para auxiliar o usuário;

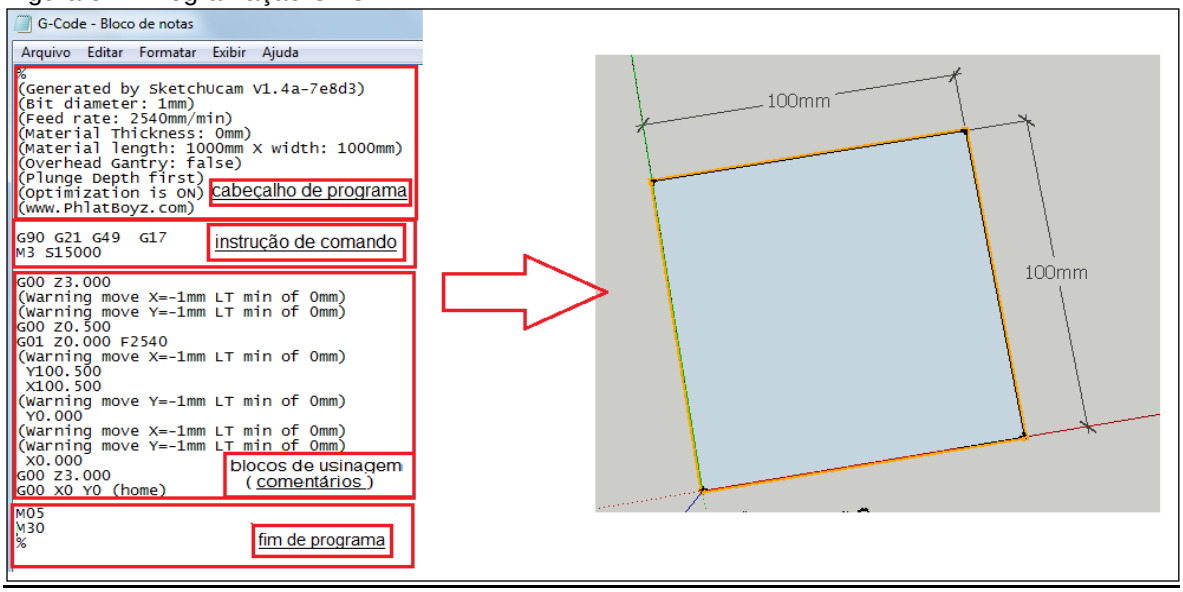
c) instrução de comando: funções ‘G’ utilizadas no programa;

d) blocos de usinagem: contém informações da trajetória de usinagem;

e) fim de programa: reposiciona os eixos em ponto inicial ou posição zero máquina.

Na figura 52 apresenta o Gcode com suas funções subdivididas de um desenho gráfico na forma quadrada.

Figura 52 - Programação CNC



Fonte: Do autor.

No bloco de usinagem se realiza o programa da trajetória de todos os eixos por etapas, a cada linha gera um movimento sobre um determinado eixo, seu endereço de posição é implementado por um valor numérico e a cada posição preparatória é seguida por uma determinada função ‘G’, no apêndice C representa os códigos e funções padrão utilizados no estudo deste projeto, que define o modo de movimentação, o tipo de interpolação e o sistema de medidas da máquina.

O movimento de posicionamento sempre ocorre da posição na qual se aproximou em último lugar para a posição de destino programada, esta posição de destino é por sua vez a posição de partida para o próximo comando de deslocamento, e assim sucessivamente, a instrução de programação implementada neste protótipo se baseia nas coordenadas absolutas (G90) e para trajetória de ferramenta sobre as coordenadas absolutas foi utilizado:

a) G00: interpolação linear com avanço rápido;

b) G01: interpolação linear com avanço programado;

c) G02 ou G03: interpolação circular com avanço programado.

9.2 ALGORITMO DE BRESENHAM PARA INTERPOLAÇÃO CIRCULAR

Visando atender a proposta deste trabalho foi implementado no código padrão G o algoritmo de Bresenham para representar trajetórias de segmento em circunferências possíveis de programar.

A geração de trajetórias circulares utilizando motores de passo é tarefa

mais complexa do que interpolar retas, além de exigir dos motores uma taxa de rotação constantemente variável, a derivada varia senoidalmente ao longo de um círculo, há geralmente grande esforço computacional no cálculo das equações que definem os pontos da trajetória, as quais envolvem multiplicações em ponto flutuante e extrações de raízes quadradas.

O estudo da programação com algoritmo de Bresenham evidenciou, basicamente, três tipos principais de abordagens para interpolações circulares:

a) a determinação dos pares ordenados de coordenadas cartesianas (X, Y) a partir da equação que define uma circunferência, $X^2+Y^2 = R^2$, onde R é o raio da circunferência;

b) divisão da circunferência em uma série de segmentos de reta;

c) algoritmos incrementais para o cálculo dos pontos da trajetória.

Com o algoritmo de Bresenham é capaz de calcular todos os pontos de uma circunferência usando apenas somas, subtrações e operações de multiplicar por dois, embora, teoricamente, este tipo de algoritmo seja o mais adequado do ponto de esforço computacional.

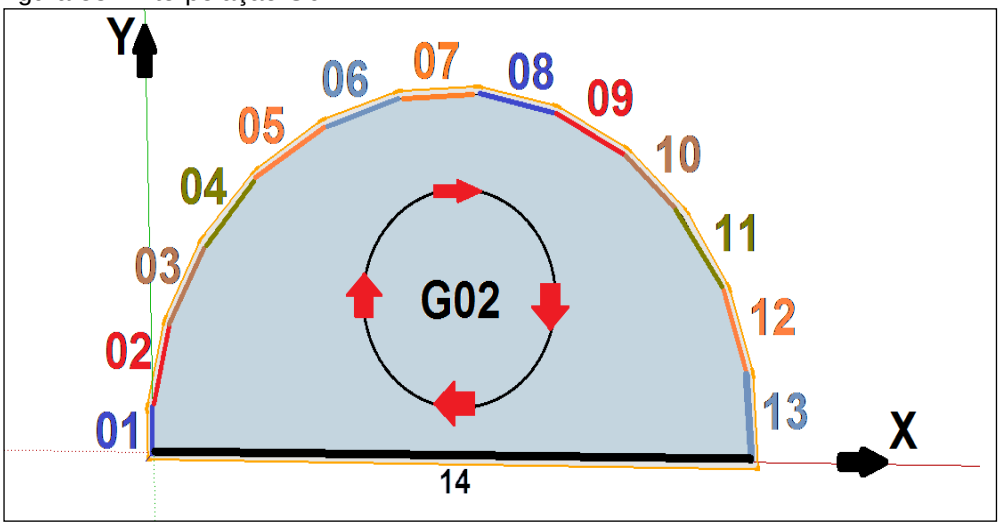
9.2.1 Interpolação circular G02 e G03

As funções G02 e G03 realiza movimentos de interpolação circular com a ferramenta sobre os sentidos dos eixos X e Y em avanço programado.

G02 – Sentido horário.
G03 – Sentido anti-horário.

Para testar o algoritmo de Bresenham foi utilizado a função G02 para interpolação circular horária da ferramenta em eixo Z sobre os eixos X e Y e criado um desenho gráfico de um raio ou semicírculo medindo 25 mm com pontos de curvatura linear enumerados, conforme figura 53.

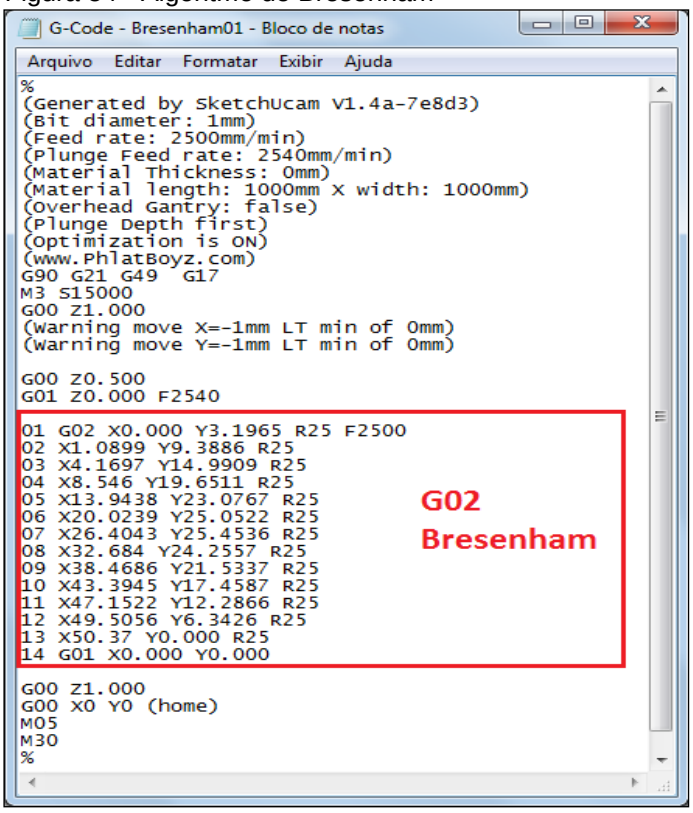
Figura 53 - Interpolação G02



Fonte: Do autor.

Cada curvatura enumerada equivale a uma linha de programa de usinagem dentro das coordenadas, para que aconteça a trajetória se deve colocar o ponto inicial do eixo X subsequente do ponto inicial do eixo Y e o ângulo referente a circunferência a ser usinada, na figura 54 apresenta as linhas do programa com o raio implementado como R25 em todas as funções até finalizar o comando.

Figura 54 - Algoritmo de Bresenham



Fonte: Do autor.

10 RESULTADOS OBTIDOS

Ao longo da construção do protótipo foi estudada uma configuração mínima de componentes de qualidade com baixo valor agregado e softwares de licença gratuita que funciona em multiplataformas de fácil acesso.

O tempo para realizar a montagem mecânica, elétrica e eletrônica deu-se em média de três meses, por conta de materiais eletrônicos que foram comprados pela internet e elementos que foram adicionados e modificados na instalação, a configuração do *hardware* e *software* levou em média de um mês, com testes efetuados foram encontrados tutoriais de melhoria para comunicar toda a máquina.

A máquina em sua estrutura de movimento linear apresenta folgas de até 0,5 mm por eixo, onde foi corrigida em suas configurações.

Este protótipo pode-se utilizar em escolas, universidades e empresas com a finalidade de ensinar a programação de comando numérico computadorizado, esta máquina permite produzir placas de circuitos eletrônicos impressos, peças de estrutura em madeira e até brinquedos pedagógicos.

11 CONCLUSÃO

O projeto atendeu todos os tópicos específicos para realização passo a passo de um equipamento físico de três eixos, que se movimenta de acordo com o pedido do usuário com ajuda de um hardware e software, plataformas open source e materiais em baixo custo.

Este protótipo abre portas para projetos tecnológicos da atualidade, fala-se muito na internet das coisas entre outros modelos de eletrônicos que geram comodidade as pessoas, o próprio usuário pode criar peças e utensílios com esta máquina que pode gerar negócios financeiros.

A máquina pode aumentar sua capacidade em área útil e com o tempo vai receber um upgrade para usinagem de peças em grande escala, como por exemplo, uma guitarra, ou seja, aprimoramentos que resultam em estudos pesados sobre processos de manufatura em comando numérico computadorizado e desenho computadorizado.

REFERÊNCIAS

BARBOZA, F. J. R.; AZEVEDO, J.; OLVEIRA, L. R; ALBUQUERQUE, M. L. **Uma Metodologia para Construção de Robôs Móveis.** Programa de Pós-Graduação em Mecatrônica, Departamento de Ciência da Computação Departamento de Engenharia Mecânica, Universidade Federal da Bahia, I Seminário Técnico-Científico, I STEC, Salvador, 2003, 7p. Disponível em: <http://www.lucianooliveira.eti.br/publicacoes_files/Ogumbot.pdf>. Acesso em: 24 abr. 2007.

BRAGA, C. N. **Os Segredos da Porta Paralela.** Revista Mecatrônica Atual, São Paulo-SP, ano 1, n.º 1, nov/2001, 32p.

BRAGA, N. C. **Eletrônica Básica para Mecatrônica.** 1.ed. São Paulo: Editora Saber, 2005, 160p.

CONRADO, R. Grbl v0.9j O que é? Para que serve? Como configurar? **Atividade Maker**, [S.l], v. online, 2018. Disponível em:<http://atividademaker.com.br/grbl- v09j>Acesso em: 20 mar.2018.

COSTA, D. D.; PEREIRA, A. G. **Desenvolvimento e Avaliação de uma Tecnologia de Baixo Custo para Programação CNC em Pequenas Empresas.** Revista Produção, Abr 2006, Vol.16, Nº1, p.48-63. Universidade Federal do Paraná, 16p. Disponível em: http://www.scielo.br/

COSTA, E. R. F. **Simulação de Movimento e Planejamento de Trajetórias para Robôs Manipuladores.** Dissertação de Mestrado, Universidade Federal de Pernambuco, Recife-PE, 2003, 74p.

COSTA, E. S; SANTOS, D. J. **Processos de Usinagem.** Centro Federal de Educação Tecnológica de Minas Gerais, Curso Técnico em Eletromecânica. Divinópolis, 2006.

FEDEL, R. **Metodologia de Desenvolvimento de Equipamento para Alimentar Deficientes Físicos.** Dissertação de Mestrado da Universidade Estadual Paulista Júlio Mesquita/Ilha Solteira, Curso de Engenharia Elétrica, São Paulo-SP, 2004, 172p.

FERREIRA, J.; MARTINS, N.; AGNELO, L.; DIAS, J. **Imagiologia Tridimensional Digital para Construção de Protótipos Industriais.** Coimbra- Portugal, Instituto de Sistemas e Robótica, Universidade de Coimbra, 2001, 11p.

FILHO, J. B. M. **Controlador Adaptativo Neural para Mesa de Coordenadas X-Y.** Curso de Pós-Graduação em Engenharia Mecânica, CT- UFPB, 2007.

FORTIN, E. **An Innovative Software Architecture to Improve Information flow from CAM to CNC.** Computers & Industrial Engineering, Vol. 46, 2004, p. 655-667.

GOELLNER, E. **Ferramenta Computacional para Acionamento de Motores de Passo Aplicados ao Projeto de Equipamentos CNC.** Dissertação de Mestrado, Santa Maria-RS, 2006, 130p.

ISO. **International Organization for Standardization.** Disponível em: http://www.iso.org. Acesso em: 31 mai. 2008.

KAIKKONEN, J.; MAKELAINEN, T.; HAKALA, H. **Advanced Design Methods and Tools for Mobile Robot Development, Intelligenc for Mechanical Systems.** Proceedings IROS, 1991, 7p.

KELLYWARE. **Manual KCAM 4 - CNC CONTROL SOFTWARE**. 2007, 31p.

KORTENKAMP, D.; BONASSO, R. P.; MURPHY, R. **Artificial Intelligence and Mobile Robots: case studies of successful robot systems**, The MIT Press, 1998.

LOPES, L. C. G. **Programando para Controle de Dispositivos pelo Computador.** Informática Industrial/Automação, SDM - Sistemas Digitais e Microprocessados CEFET-MG Campus III - Uned Leopoldina, 2007, 36p.

LYNCH, M. **Computer Numerical Control: Acessory Devices**. New York: McGraw-Hill, 1994, 262p.

LYNCH, M. **The Key Concepts of CNC.** Modern Machine Shop, Cincinnati, Vol. 69, Nº 11A, 1997, p.81-144.

MACH3. **Manual de Utilização do Software Mach3.** 2005, 43p.

MACHADO, A. **Comando Numérico Aplicado às Máquinas-Ferramenta.** 4ª ed., São Paulo, Editora Ícone, 1990, 312 p.

NBR. **Associação Brasileira de Normas Técnicas.** Disponível em: http://www.abnt.org.br. Acesso em: 31 mai. 2008.

PAZOS, F. A; LOVISOLO, L. **Automação de Sistemas e Robótica.** 1 ed. Rio de Janeiro: Editora Axcel Books do Brasil, 2002, 384p.

PENTEADO, F. A. C. A. **Processo de Usinagem dos Metais.** Revista CADware(r) Publishing Brazil. Agosto/2002.

PERACETTA, L. F.; UNIANDRADE, A. R. **Sistema de Aquisição de Dados e Controle de Processos Através da Porta Paralela.** Simpósio, 2003, 10p. Disponível em: http://www.uniandrade.br/simposio/pdf/comp105.pdf. Acesso em: 23 set. 2008.

PEREIRA, A. G. **Desenvolvimento e Avaliação de um Editor para Programação CN em Centros de Usinagem.** Dissertação de Mestrado, Curitiba-PR, 2003, 122p.

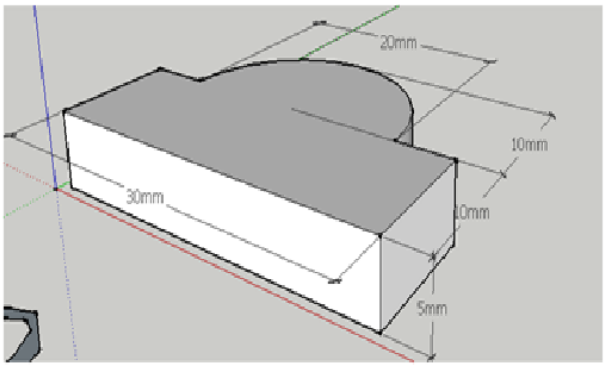
APÊNDICE(S)

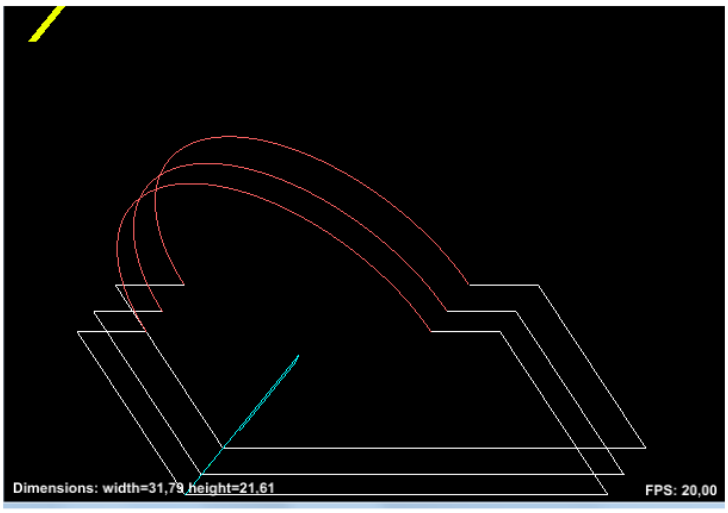
APÊNDICE A – Investimento Construção protótipo

Produto	Qtdade	Val/Unitário	Total por peça
Barra roscada de 6 mm	3	R$ 3.00	R$ 9.00
Madeira MDF	1	R$ 80.00	R$ 80.00
Rolamento Corrediça	6	R$ 14.00	R$ 84.00
Porca	3	R$ 1.00	R$ 3.00
Parafusos para madeira	20	R$ 0.45	R$ 9.00
Acoplamento	3	R$ 30.00	R$ 90.00
CNC Shield V3	1	R$ 250.00	R$ 250.00
Arduino UNO	1	R$ 70.00	R$ 70.00
Driver A4988	3	R$ 30.00	R$ 90.00
Fonte ATX (usada)	1	R$ 30.00	R$ 30.00
Micro Retifíca	1	R$ 230.00	R$ 230.00
Fresa	1	R$ 100.00	R$ 100.00
TOTAL	**44**	**R$ 838.45**	**R$ 1,045.00**

APÊNDICE B – Linguagem CNC / G-Code

```
%
(Generated by SketchUcam V1.4a-7e8d3)
(Bit diameter: 1mm)
(Feed rate: 240mm/min)
(Plunge Feed rate: 350mm/min)
(Material Thickness: 5mm)
(Material length: 200mm X width: 150mm)
(Overhead Gantry: false)
(Multipass enabled, Depth = 2mm)
(Plunge Depth first)
(Optimization is ON)
(www.PhlatBoyz.com)
(Loaded profile teste2)
G90 G21 G49  G17
M3 S15000
G00 Z5.000
(Pass: 1)
 X0.786 Y0.607
G00 Z0.500
G01 Z-2.000 F350
 Y11.607 F240
 X5.847
G02 X6.1392 Y13.8261 R10.5
 X7.1885 Y16.3595 R10.5
 X8.8579 Y18.535 R10.5
 X11.0334 Y20.2043 R10.5
 X13.5668 Y21.2537 R10.5
 X16.2855 Y21.6117 R10.5
 X19.0043 Y21.2537 R10.5
 X21.5377 Y20.2043 R10.5
 X23.7132 Y18.535 R10.5
 X25.3825 Y16.3595 R10.5
 X26.4319 Y13.8261 R10.5
 X26.724 Y11.6073 R10.5
G01 X31.786
 Y0.607
 X0.786
(Pass: 2)
 Z-4.000 F350
 Y11.607 F240
 X5.847
G02 X6.1392 Y13.8261 R10.5
 X7.1885 Y16.3595 R10.5
 X8.8579 Y18.535 R10.5
 X11.0334 Y20.2043 R10.5
 X13.5668 Y21.2537 R10.5
 X16.2855 Y21.6117 R10.5
 X19.0043 Y21.2537 R10.5
 X21.5377 Y20.2043 R10.5
 X23.7132 Y18.535 R10.5
 X25.3825 Y16.3595 R10.5
 X26.4319 Y13.8261 R10.5
 X26.724 Y11.6073 R10.5
G01 X31.786
 Y0.607
 X0.786
(Pass: 3)
 Z-5.500 F350
 Y11.607 F240
 X5.847
G02 X6.1392 Y13.8261 R10.5
 X7.1885 Y16.3595 R10.5
 X8.8579 Y18.535 R10.5
 X11.0334 Y20.2043 R10.5
 X13.5668 Y21.2537 R10.5
 X16.2855 Y21.6117 R10.5
 X19.0043 Y21.2537 R10.5
 X21.5377 Y20.2043 R10.5
 X23.7132 Y18.535 R10.5
```





APÊNDICE C – Funções G

G00 Interpolação linear em avanço em rápido

G01 Interpolação linear com avanço programado

G02 Interpolação circular no sentido horário

G03 Interpolação circular no sentido anti-horário

G04 Permite uma parada num tempo programado:

G17 Define o plano de trabalho XY (Valor padrão)

G18 Define o plano de trabalho XZ

G19 Define o plano de trabalho YZ

G20 Programação em polegadas

G21 Programação em milímetros

G40 Cancela compensação da ferramenta

G41 Faz compensação do raio da ferramenta à esquerda da trajetória programada:

G42 Faz compensação do raio da ferramenta à direita da trajetória programada:

G53 Cancelamento dos deslocamentos de origem - Ponto zero máquina

G54 1.º Deslocamento de origem - Ponto zero peça

G55 2.º Deslocamento de origem - Ponto zero peça

G56 3.º Deslocamento de origem - Ponto zero peça

G57 4.º Deslocamento de origem - Ponto zero peça

G59 Deslocamento de origem aditivo externo

G60 Parada precisa

G64 Deslocamento contínuo

G70 Sistema de medidas em polegadas

G71 Sistema de medidas métrico

G80 Cancelamento de ciclo fixo

G81 Ciclo de furação simples

G83 Ciclo de furação tipo pica-pau

G84 Ciclo de roscamento com macho

G85 Ciclo de alargamento

G86 Ciclo de mandrilamento

G90 Programação em sistemas de coordenadas absolutas

G91 Programação em sistemas de coordenadas incrementais

G94 O avanço é programado em mm/min

G95 O avanço é programado em mm/rot

G97 A rotação é programada em RPM

APÊNDICE D – Artigo

DESENVOLVIMENTO DE UMA MÁQUINA CNC COM HARDWARE E SOFTWARE OPEN SOURCE

Diorgines M. Machado[1], Sérgio Coral[2]
[1] Acadêmico do Curso de Ciência da Computação – Universidade do Extremo Sul Catarinense
[2] Professor do Curso de Ciência da Computação – Universidade do Extremo Sul Catarinense
diorginesmattos@hotmail.com, sergiocoral@unesc.net

Abstract. *Development of a prototype, industrial model, of a computerized numerical control machine for machining operations using Arduino UNO and CNC Shield communication hardware with GRBL firmware implemented according to configuration of the control points for the realization of the X, Y and Z linear axis movements adapted by Nema 17 step motors with the help of computer graphics software by Sketchup where it determines the lines of segment and geometric section of each axis through the Sketch UCAM extension, tool that transforms drawing in CNC Language based on the fundamentals of the algorithm of Bresenham hardware motion interpolation in standard G-CODE language file interpreted by Universal G-Code Sender software to perform machining operations..*

Resumo. *Desenvolvimento de um protótipo, modelo industrial, de uma máquina com comando numérico computadorizado para operações de usinagem utilizando hardware de comunicação Arduino UNO e CNC Shield com firmware GRBL implementado conforme configuração dos pontos de controle para realização dos movimentos nos eixos lineares X, Y e Z adaptados por motores de passo Nema 17, com o auxílio da computação gráfica pelo software Sketchup onde determina as linhas de segmento e secção geométrica de cada eixo através da extensão Sketch UCAM, ferramenta que transforma desenho em Linguagem CNC baseado nos fundamentos do algoritmo de Bresenham para interpolação de movimento por hardware em arquivo padrão de linguagem G-CODE interpretado pelo software Universal G-Code Sender para a realizar operações de usinagem.*

1. Introdução

Desenvolver uma máquina com comando numérico computadorizado com plataforma de hardware de fácil acesso e softwares gratuitos com programação livre era uma tarefa difícil, hoje a automação está participando das pequenas oficinas e laboratórios com recursos e orientações que se traz através de sites de cientistas motivados a inovação, trocando seu espaço de conhecimento para toda rede interessada a invenções.

A indústria vive nos dias atuais "a era das máquinas livres" incentiva a adesão de novos inventores que revoluciona o uso de plataformas open source, que neste protótipo, os hardwares trazem vida em movimento sincronizado seccionado por eixos obedecidos por comandos através do software de programação (Goodrich, 2012).

O Arduino é um hardware que vem transformando o mercado do conhecimento, com bibliotecas já instaladas e com funções eletrônicas programadas através de uma IDE, pode-se gerar pulsos de movimentos elétricos e até mecânicos, com objetivo de auxílio ou transformação como por exemplo serviços de usinagem, para Pearce a plataforma Arduino torna simples a tarefa em realizar processos automatizados (Pearce, 2012).

O presente trabalho apresenta um estudo sobre o uso do Arduino em operações de usinagem com um sistema programado adaptado sobre uma máquina, detalhando a teoria do projeto

dividido em pontos de construção mecânica, elétrica, eletrônica e configuração de hardware e software com metodologia utilizada e por fim uma análise dos resultados obtidos do protótipo.

2. Programação CNC

Os movimentos fazem parte de uma máquina CNC, movimentos lineares ou rotação, cada movimento é parte de um eixo associando em letras X, Y e Z (PEREIRA, 2006).

A programação pode ser de forma manual ou automatizado, classificada especificamente em programação manual ou programação em CAD e CAM (PEREIRA, 2006).

A programação manual é indicada a trabalhos de fresamento, a programação é feita sem utilização de recursos computacionais, são documentadas em manuscrito do programa, que são dados de listagem de posições da ferramenta em relação à peça a ser usinada, onde a máquina precisa seguir para executar o processo de transformação (PEREIRA, 2006).

A programação CAD e CAM são softwares onde o sistema CAD é um software gráfico computadorizado para criar uma peça graficamente, e o CAM é um sistema onde o software faz a leitura de cada margem do desenho em esboço e transforma em linha de programa, este processo de programação é o mais comum nas indústrias de usinagem de peças seriadas (PEREIRA, 2006).

A criação de um programa CNC por CAD e CAM passa pelas etapas de edição, simulação e transmissão do código para a máquina para ser compilado, onde o editor obtém de ferramentas gráficas, intervindo a erros no processo pelo motivo de ver a peça dentro das suas dimensões saber quais funções será realizado para sua criação, a simulação baseia-se na representação gráfica da trajetória das ferramentas em um plano escolhido, a transferência do código para a máquina tem como objetivo a redução do tempo, quando comparado à introdução via teclado da máquina, e a eliminação de eventuais erros de digitação, a mesma é feita pela porta serial RS232. Os parâmetros de comunicação, tais como velocidade, paridade, tamanho de palavra, bits de parada, modo de transferência e o número da porta serial devem ser previamente definidos pelo usuário e o arquivo a ser transmitido deve conter apenas caracteres ASCII e possuir tamanho compatível com a memória disponível na máquina (PEREIRA, 2006).

A tecnologia de baixo custo proposta foi implementada com o intuito de reduzir o tempo de programação e, reduzir o custo dos processos de usinagem em ambientes fabris. A avaliação foi realizada em três etapas: seleção de empresas que atendessem a um determinado perfil, treinamento e avaliação de técnicos dessas empresas e teste do protótipo no chão de fábrica. Este sistema, embora não possa ser considerado um sistema CAD e CAM (o usuário digita as coordenadas para gerar a trajetória da ferramenta), quando comparado àqueles observados em empresas que ainda fazem uso da programação manual, representa uma alternativa viável para redução do tempo total de usinagem e consequentemente de custos de fabricação. (PEREIRA, 2006).

3. Algoritmo de Bresenham

É relativamente fácil criar um algoritmo para traçar retas e gerar pontos aproximados sobre bases matemáticas de pontos flutuantes. Hoje possivelmente as ULA’s dos computadores possuem instruções para trabalhar com este tipo de sistema flutuante de forma rápida, porém no caso de microcontroladores esta situação se converte em problema devido à baixa velocidade de operação e as reduzidas instruções (MORO, 2009).

O uso dos microcontroladores na automação é um problema se não usados corretamente. Usar ponto flutuante exige do processador vários ciclos de clock acarretando na defasagem da comparação e controle no processo automatizado. Para contornar esta situação opta-se por

algoritmos de alto nível minimizando o trabalho computacional. Em 1965 Jack E. Bresenham, então funcionário da IBM propõe um algoritmo eficaz para a época que calculava melhor ponto nas trajetórias retilíneas por meio de lógica com números inteiros (MORO, 2009).

Foi proposto por Bresenham em 1965 um algoritmo clássico que utiliza apenas variáveis inteiras, Algoritmo de Bresenham ou conhecida no mercado da computação gráfica como Algoritmo do Ponto-Médio, o algoritmo permite fazer os cálculos de (xi+1, yi+1) incrementalmente, usando os cálculos já feitos para (xi, yi), o algoritmo assume que a inclinação está entre 0 e 1 no 1º octante (outras inclinações podem ser usadas por simetria), e o ponto (x1, y1) é o inferior esquerdo, e (x2, y2) é o superior direito. Partindo do ponto P(x1, y1), a escolha do próximo ponto a ser desenhado é analisando o ponto médio M, sendo que, se M está acima da reta, o próximo ponto a ser desenhado é o da direita (E), e se M estiver abaixo da reta, o próximo a ser desenhado é o ponto acima e a direita (NE), (FERREIRA, 2004).

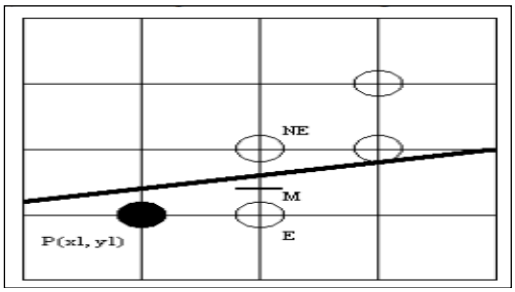


Figura 1. Reta

Utilizando esta função é possível analisar que P(x,y) é zero para pontos sobre da reta, negativa para pontos acima e positiva abaixo. Logo em seguida é analisado o ponto médio M com base ao pixel atual P e verificado qual pixel foi escolhido, ME ou M. Uma nova comparação lógica é determinada para saber o quanto e qual valor de x e y será incrementado para o próximo pixel dependendo do pixel atualmente escolhido (BRESENHAM, 1965, tradução nossa).

Bresenham (1965, tradução nossa) por meio dos cálculos e eliminando a fração da equação primitiva da reta encontra equações de decisões, ou métodos que satisfazem o coeficiente angular caso as seleções dos pixels sejam diferentes.

Por fim o Bresenham (1965, tradução nossa) apresenta condições ainda de recursão, caso a inclinação da reta seja invertida, x1 > x2, para satisfazer o algoritmo por completo. O algoritmo de Bresenham pode ser desenvolvido em qualquer linguagem, Figura 2, e em qualquer sistema de modo eficaz, por possuir lógica inteira.

```
/** Desenha uma reta entre dois pontos **/
void DrawLine(point p0, point p1, color color){
        int slope;
        int dx, dy, incE, incNE, d, x, y;

        // Onde inverte a linha x1 > x2
        if (p0.x > p1.x){
            DrawLine(p1, p0, color);
            return;
        }

        dx = p1.x - p0.x;
        dy = p1.y - p0.y;

        slope = (dy < 0) ? -1 : 1;
        if (dy < 0){
            dy = -dy;
        }

        // Constante de Bresenham
        incE = 2 * dy;
        incNE = 2 * dy - 2 * dx;

        d = 2 * dy - dx;
        y = p0.y;

        for (x = p0.x; x <= p1.x; x++){
            PutPixel(x, y, color);
            d += (d <= 0)? incE: incNE;
            if(d > 0) y += slope;
        }
}
```

Figura 2. Algoritmo de Bresenham em C

Para construção de circunferências utiliza-se o Algoritmo do Ponto-Médio para Circunferências, uma equação com centro na origem e raio R, no plano cartesiano é dada por: $x^2 + y^2 = R^2$, seja a função $F(x, y) = x^2 + y^2 - R^2 = 0$, o valor 0 é sobre a circunferência, positivo fora dela e negativo dentro, considerando um arco de 45º, onde o ponto inicial é dado por P(x, y) e sendo (x = 0) e (y = R), a escolha do próximo ponto a ser desenhado é analisando o ponto médio M, sendo que, se M está dentro da circunferência, o ponto escolhido é o próximo a direita (E), e se M está fora ou sobre a circunferência, o ponto escolhido é o próximo a direita e abaixo (SE), (FERREIRA, 2004).

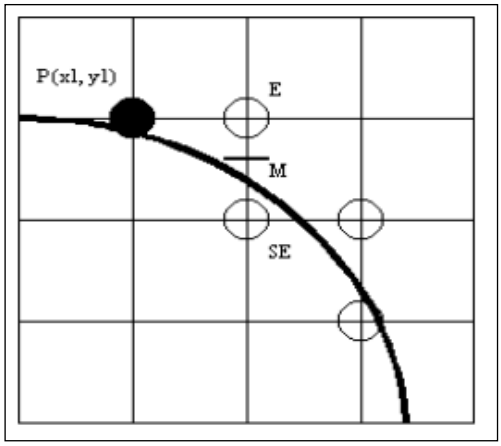


Figura 3. Circunferência

Para construção de elipses, utiliza-se Algoritmo do Ponto-Médio para Elipses, a equação da elipse centrada em (0, 0) é dada por: $F(x, y) = b^2x^2 + a^2y^2 - a^2b^2 = 0$, onde 2a é o comprimento do eixo maior (eixo x) e 2b é o comprimento do eixo menor (eixo y), como no traçado da elipse ocorre mudança na inclinação, divide-se o 1º quadrante em duas regiões, sendo, o limite entre as duas regiões é o ponto da curva cuja tangente tem inclinação igual a –1, assim, enquanto $a^2y > b^2x$ permanece-se na região 1, caso contrário, muda-se para a região 2 (FERREIRA, 2004).

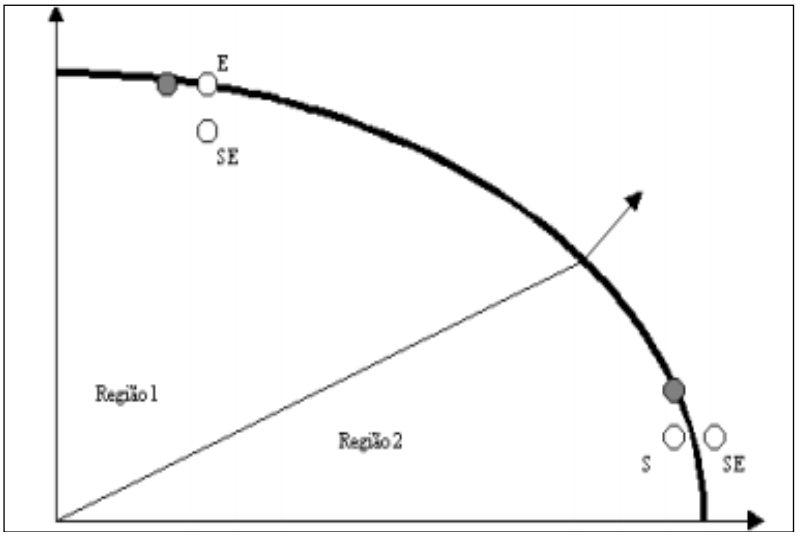


Figura 4. Inclinação

Seja a função $F(x, y) = b^2x^2 + a^2y^2 - a^2b^2 = 0$, o valor 0 é sobre a elipse, positivo fora dela e negativo dentro, considerando um arco de 45º, onde o ponto inicial é dado por P(x, y) e sendo (x = 0) e (y = b), a escolha do próximo ponto a ser desenhado é analisando o ponto médio M, na região 1, se M está dentro da elipse, o ponto escolhido é o próximo a direita (E), e se M está fora ou sobre a elipse, o ponto escolhido é o próximo a direita e abaixo (SE), na região 2, se M está dentro da elipse, escolhe-se (SE), e se M está fora ou sobre a elipse, escolhe-se (S), como

os incrementos são calculados para o 1º e 2º octantes, deve-se usar simetria de ordem 4 para desenhar os pontos dos demais octantes, formando a elipse (FERREIRA, 2004).

Em um sistema automatizado a última área consiste no software gerenciador de todo sistema. Este software entra como analisador das informações geradas por todo processo automatizado, por meio de comunicações entre o sistema de controle. O Arduino por possuir maior abstração facilita no processo de comunicação entre um computador e o sistema. Esta comunicação deve ser elaborada para receber os dados de um software e converte-los em passos para os motores. Protocolos devem garantir à exatidão no recebimento das informações enviadas a máquina, de tal maneira que comandos e dados sejam interpretados pelo Arduino (BURIGO, 2014).

4 Construção protótipo máquina CNC

Um protótipo CNC ao construir engloba em todo seu projeto uma instalação mecânica, elétrica e eletrônica, não se considera cálculos de complexidade sobre sua resistência estrutural, pelo motivo que se trata de um equipamento em menor escala, apenas considera o objetivo de complemento ao ensino de programação em máquinas automatizadas com sistema seriado.

4.1 Mecânica

Para construir um protótipo máquina CNC no primeiro ponto determina-se sua área útil de trabalho e o tipo de material produzido ou usinado e o segundo ponto é a característica da base padrão dos eixos de qual forma será alojado, dentro destes requisitos o protótipo obtém estrutura mecânica e corpo visual de uma máquina CNC.

4.1.1 Projeto máquina estrutural

O protótipo da máquina CNC deste trabalho seguiu o modelo *Router*, que é muito utilizado em usinagem de materiais maleáveis como madeiras e plásticos, a máquina opera com três eixos direcionais, sabendo que o eixo X responde a direção longitudinal, eixo Y responde a direção transversal e eixo Z responde a direção vertical, estes eixos trabalham independente em suas posições de base (figura 5).

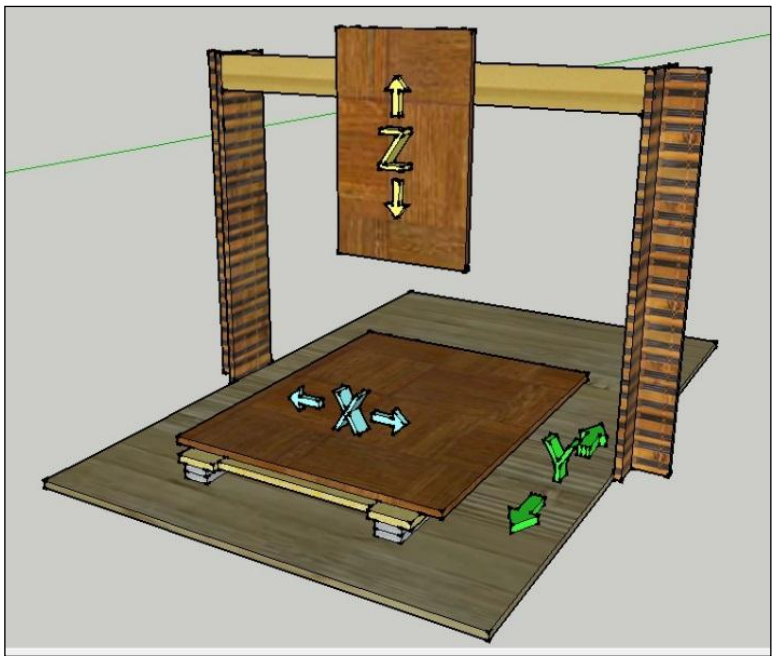


Figura 5. Projeto protótipo CNC

4.2 Elétrica e eletrônica

Instala-se a eletroeletrônica de uma máquina CNC por partes, sendo módulos da placa controladora, módulo dos drivers de potência, módulo dos atuadores, fonte de energia e ferramenta elétrica ou eixo árvore.

4.2.1 Placas de controle

Uma placa de controle é responsável pela comunicação de dados da máquina, interpreta os dados a ser enviados pelo computador, a placa de controle coordena os movimentos executados pelos eixos sendo a aceleração e acionamento do motor de passo conforme desejável.

4.2.1.1 Arduino UNO

A placa Arduino UNO integra com as configurações de *firmware* da *Grbl*, para acessar as portas e receber dados para à máquina com amplo suporte de desenvolvimento, possui entrada USB e placas de circuitos que podem se conectar para toda ampliação eletrônica, no caso do protótipo CNC, o Arduino UNO conecta a uma placa CNC *Shield*, específica em se comunicar com os drivers de potência dos motores de passo e toda alimentação elétrica do protótipo.

4.2.1.2 CNC Shield

No Arduino UNO pode-se conectar através de fios em direto aos motores de passo ou elemento eletrônico do protótipo, sendo que os *drivers* de potência ou botões ou até mesmo chaves de liga e desliga, no entanto, se adapta a esta ligação de forma incomplexa e sucessível a erros de comunicação (figura 6).

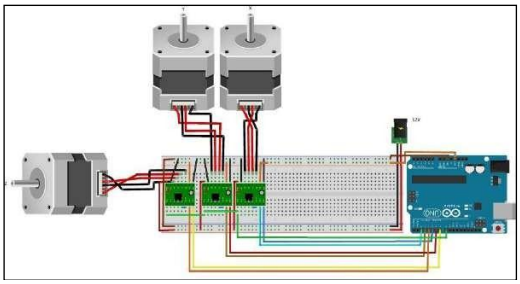


Figura 6. Arduino UNO ligados aos drivers de potência

Por motivo de organizar os elementos eletrônicos e elétricos conectados, utiliza-se uma placa pronta junto ao Arduino UNO que se chama CNC S*hield*, a fim de evitar falhas na comunicação de dados.

A CNC *Shield* é uma placa de circuitos de fácil conexão com a placa Arduino UNO ou com outra placa CNC S*hield*, quando conecta ao Arduino o mesmo expande sua capacidade em integrar *displays*, cartões de memória ou módulos como por exemplo o *bluetooth*.

Neste protótipo se utiliza o Arduino CNC *Shield V3* (figura 7), para conectar aos *drivers* de potência, Arduino UNO, motores de passos e fonte de energia.

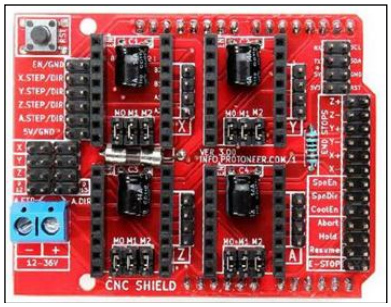


Figura 7. Arduino CNC Shield V3

4.2.1.3 Drivers de potência

Os *drivers* de potência têm a função em transformar sinais elétricos de baixa potência por sinais de alta potência, dentro dos limites de tensão e corrente, sendo necessário o uso da alimentação no dispositivo maior que a nominal do sistema que se controla.

No protótipo o *driver* de potência é responsável por ligar os motores de passo, o *driver* ideal que se utiliza é o *A4988 Stepper Motor Driver* (figura 8).

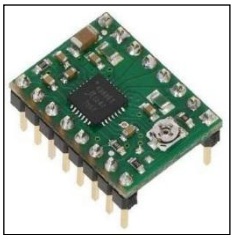


Figura 8. A4988 Stepper Motor Driver

O *driver* controla os motores de passo e pode trabalhar entre tensões de 8 e 35 volts, sendo assim, entrega até 1,5 Ampères por bobina, na figura 9 representa o esquema elétrico de ligação.

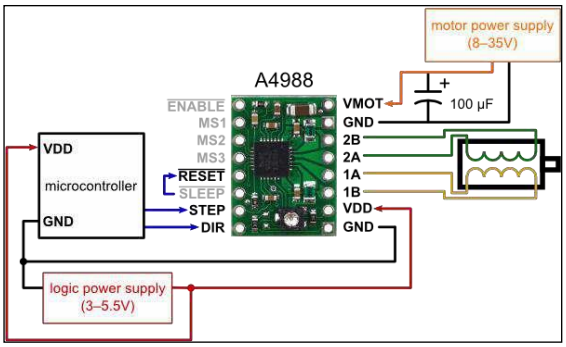


Figura 9. Esquema elétrico de ligação

O esquema eletrônico é simples e prático entre o *driver* e CNC *Shield* pois já tem posição de encaixe própria para o *driver A4988*, de acordo com a figura 10, obtendo assim a placa principal conjunta do protótipo.

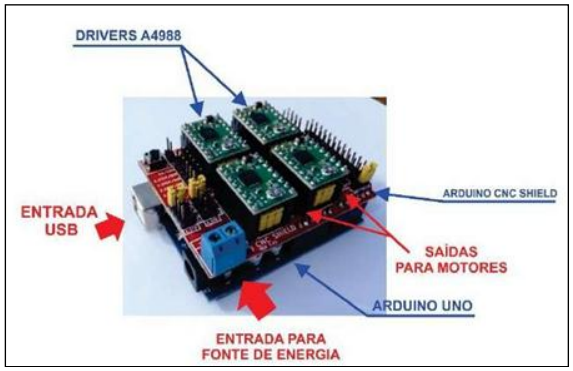


Figura 10. Montagem driver

Usa-se o conector de alimentação da fonte *ATX* para energizar a placa CNC *Shield V3*, sabe-se que as cores dos fios indicam suas polaridades, o fio de cor amarela indica o pólo positivo e o fio na cor preta indica o pólo negativo, deste modo, conecta-se os fios na alimentação da placa CNC *Shield V3* (figura 11).

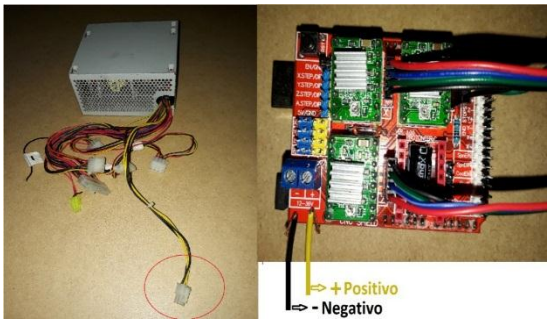


Figura 11. Energizar placa CNC Shield V3

Como se trata de alimentação em baixa corrente, não é necessário neste protótipo de refrigeração com ventilação na placa CNC *Shield V3*.

4.3 Motores de passo

Conhecido como atuadores os motores de passos têm o princípio de converter um dado tipo de energia em sentido de giro, no caso deste protótipo, o sentido de giro se torna em sentido de movimento linear pelos eixos X, Y ou Z.

Os motores de passo utilizado neste protótipo é um *Nema 17* consome 1,5 Ampères por atuador e dimensão ideal para o projeto (figura 12).

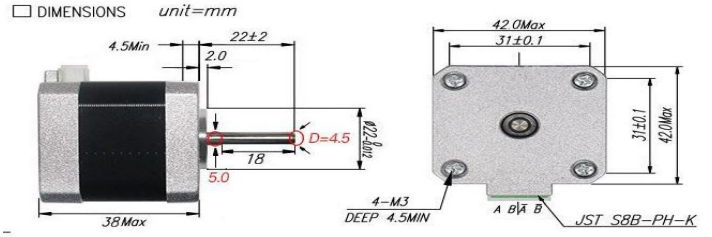


Figura 12. Dimensionamento motor de passo Nema 17

Os *drivers* de potência devem controlar todo fluxo de energia em cada fase do motor de passo deste projeto, a capacidade energética se calcula conforme a corrente e tensão nominal do atuador, os motores de passo que se utiliza têm dimensionamento padrão *Nema 17*, com 39 N.cm (Newton por centímetro) de torque, passo de 1.8 graus e corrente máxima de 1,5 Ampères por fase.

Com os dados de referência elétricos dos fabricantes do motor de passo Nema 17 e do driver A4988, através da fórmula do fabricante do driver (figura 13), regula-se a tensão de corrente contínua nominal ideal de trabalho resultante a 1200 mV, com multímetro.

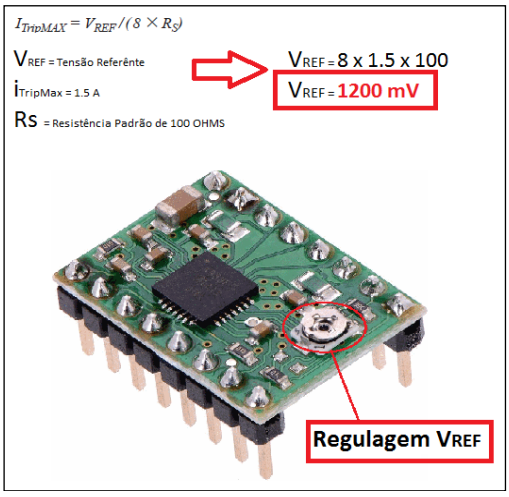


Figura 13. Fórmula e regulagem do driver

No protótipo o motor de passo deve-se instalar na base da máquina e encaixar no extremo da barra roscada onde consta o acoplamento, fixo com suporte na carcaça do motor para não mover no momento de operação, a alimentação elétrica parte da CNC *Shield V3* com cabo de 4 vias conforme a figura 14.

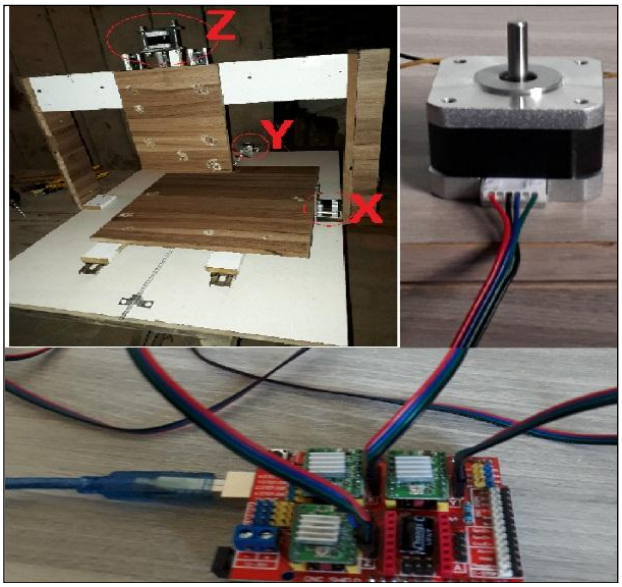


Figura 14. Instalação elétrica motor de passo

5 Softwares e linguagem CNC

O protótipo CNC se comunica via cabo ligado ao Arduino Uno e computador, e para realizar os parâmetros de linguagem onde o Hardware irá obedecer e deslocar, usa-se o software Arduino para programar a CNC Shield V3 e para desenhar peça que será feita com função CAD usa-se o software SketchUp 2016 com biblioteca de ferramentas CAM adicionadas para gerar a linguagem CNC, conhecida no mundo de máquinas *Router* como G-Code e seu Software de leitura da linguagem CNC G-Code é o Universal Gcode Sender, onde têm o objetivo principal em dar coordenadas em medidas precisas para os motores de passo de cada eixo, formando assim a peça mensurada em desenho intangível em uma peça real ou tangível.

Todas as plataformas utilizadas neste protótipo são livres ou open source, encontra-se em sites de busca ou sites de máquinas caseiras com Arduino UNO, exige apenas entendimento e conhecimento no mercado de máquinas operatrizes CNC, softwares CAD e CAM e programação eletrônica.

5.1 Software Arduino

Como usa-se o hardware Arduino UNO, o software Arduino é que opera e programa sobre esta placa, um software de aplicação em multiplataforma escrita em Java com recursos de sintaxe, parênteses e identificação automática, compila e carrega os programas desenvolvidos para a placa de forma fácil, algumas bibliotecas dão a capacidade de programar em C ou C++, ambiente requer duas seções para funcionamento da programação na placa Arduino UNO, *VOID SETUP*, são as configurações que inicia os componentes eletrônicos que irá operar no programa e *VOID LOOP*, faz repetir o comando programado no bloco até que o usuário finalize a seção ou desliga.

5.2 Firmware GRBL

O firmware *GRBL* é um código aberto desenvolvido em C, onde requer um *hardware* para operar máquinas CNC ou impressora 3D já neste protótipo usa-se Arduino UNO com a placa CNC Shield V3 para se movimentar os eixos da CNC.

5.2.1 Configuração firmware GRBL

É necessário informar ao programa uma série de parâmetros físicos e de configuração da máquina CNC para que o controle seja feito corretamente, a estratégia aplicada de maneira geral em *softwares* comerciais é a utilização de arquivos de configuração, que não obrigam o usuário a informar suas preferências a cada nova execução.

Para configurar abrir o arquivo *config* em bloco de notas, selecionar a linha *247-#define VARIABLE_SPINDLE // Default enabled. Comment to disable*, esta linha do programa define a variação de velocidade do eixo árvore, este eixo não foi instalado no protótipo, para isto precisa-se desabilitar esta função, comenta-se a linha inteira ou digitar duas vezes o caractere '// '.

A leitura de informações do arquivo *config.txt* é feita sempre que o programa se inicia, há opções para impressão dos valores lidos na tela e para realização de nova leitura a qualquer momento durante a execução, assim salvar o arquivo *config* alterado na linha do programa, abrir o software Arduino, na aba entrar em *Sketch*, *Incluir Biblioteca* e *Adicionar Biblioteca.ZIP*, encontrar a pasta *grbl* no diretório e abrir, este processo irá instalar a biblioteca *GRBL* e compilar para o hardware Arduino.

5.3 Cad e Cam

Para criar uma peça a ser usinado no protótipo utiliza-se ferramentas para dimensionar e formar em modelo de desenho técnico com suas medidas reais e tornar estas medidas dimensionadas em linguagem programada CNC, estes softwares chamam-se Computer Aided Design e Computer Aided Manufacturing ou abrevia-se CAD e CAM, modelo que se utiliza no mercado industrial principalmente nos setores de Engenharia em Processos.

O software CAD que se utiliza neste protótipo para criar as peças é o Sketchup 2016, além se tratar de um software gratuito também é fácil de operar, ferramentas clássicas para desenho computadorizado como o lápis digital, linhas, largura, espessura e figuras rápidas como círculo e quadrado em formatos de objetos 2D e 3D e também instalar bibliotecas para implementar funções extras nos projetos.

Para se movimentar e manufaturar peça no protótipo máquina CNC usa-se o software CAM gratuito *Universal Gcode Sender V 1.0.9*, cria-se todo o processo de produção dentro do que se configurou entre software e hardware, e com base na programação CNC, inicia-se o deslocamento de acordo com o que se dimensionou em CAD, pode-se simular a trajetória de usinagem operatriz antes mesmo de usinar e também operar em modo manual em eixos independentes.

5.3.1 Configuração Sketchup

O software Scketchup para funcionar no protótipo deve-se instalar a extensão adicional SketchUCAM que se encontra no site da *OPENBUILDS*, para instalar tem que acessar a opção *janela*, *instalar extensão* em *preferências do sistema*, encontrar o diretório do arquivo e abrir *SketchUCAM*, neste modo se adiciona novas ferramentas na área de trabalho do software.

Na extensão SketchUCAM se ajusta os parâmetros de manufatura do desenho com a máquina, sendo que o desenho peça tem-se que se usinar dentro do limite espacial de área do protótipo, abrir a ferramenta *phlatboyz parameters* e configurar, todas as funções usais:

n) *Feed Rate*, velocidade de corte entre X e Y em milímetros por segundo;

o) *Plunge Rate*, velocidade do eixo Z com a mesa em milímetros por segundo;

p) *Material Thickness*, espessura do material bruto em milímetros;

q) *In/OutsideOvercut*, porcentagem de corte configurar mediante folgas no eixo Z;

r) *Bit Diameter*, diâmetro da fresa em milímetros;

s) *TabWidth*, espessura do material que fica entre a peça e o material bruto na largura em milímetros;

t) *TabDepth*, espessura do material que fica entre a peça e o material bruto na espessura em por cento;

u) *Safe Travel Z*, subir eixo em relação ao ponto zero máquina em milímetros;

v) *Safe Lenght X*, dimensão de curso em X em 150 milímetros;

w) *Safe Width Y*, dimensão de curso em Y em milímetros;

x) *Step Over*, material que a fresa não corta por passada manter o padrão em porcento;

y) *GenerateMultipass*, gerar passos de usinagem;

z) *MultipassDepth*, espessura para usinar a cada passada em milímetros.

As peças são dimensionadas em milímetros (mm) e com o desenho da peça pronta deve-se criar o G-Code das linhas de medição de cada aresta do desenho, seleciona-se a ferramenta *Phlatboyz Outside Cut Tool*, define-se as linhas de usinagem com a ferramenta de seleção e na opção *Generate Phlatboyz GCode* gera-se a coordenada de linguagem CNC ou G-Code para leitura e operar a máquina.

5.3.2 Configuração Gcode Sender

Para acessar ou configurar funções no software *Universal GcondeSender* tem-se que conectar com o hardware *Arduino UNO* via cabo *USB* e placa *CNC SHIELD V3* ligado na *Fonte ATX*, no software na opção *CONNECTION*,selecionar a porta *COM* (*USB*), identificar em *BAUD* a velocidade que deve se comunicar com o firmware *GRBL*, clicar no botão *OPEN* na aba *CONSOLE* onde deve-se aparecer a mensagem *Grbl 0.9j ['$' for help]* , comando conectado com o *Arduino UNO*.

Com o hardware conectado pode-se configurar o software Universal Gcode Sender com os parâmetros característicos do protótipo para isto deve-se ir na aba *SETTINGS* acessar *FIRMWARE SETTINGS* e abrir a opção *GRBL*.

Na janela de configurações *GRBL*, altera-se os valores em *VALUE*, seguidos de sua ordem em *SETTING* conforme cada opção em *DESCRIPTION*.

Na aba de operação entrar em *FILE MODE* e acessar o *BROWSE*, encontrar o diretório do arquivo com extensão em *G-Code* salvo pelo *Scketchup*,

No protótipo cada eixo se posiciona em ponto zero ou ponto inicial, com o arquivo G-Code aberto no software, pode-se visualizar simulação gráfica de todo o processo de usinagem e as linhas da linguagem de programação, em *SEND* inicia-se o processo de conformação da peça.

A máquina finaliza seu processo de usinagem e retorna automaticamente para seu ponto zero, o resultado é uma peça pronta com suas dimensões mensuradas.

6 Programação CNC

Para geração de dados de programação no protótipo é gerado através da extensão Sketch UCAM no software Sketchup, onde transforma linhas de medidas do desenho em comando de programa em formato de texto obedecendo o sistema de código padrão ‘G’ ou G-Code, este programa determina as coordenadas cartesianas com diversas colunas de pares ordenados que permitem

ao usuário, com o auxílio do programa Universal Gcode Sender, visualizar graficamente as trajetórias calculadas pelo programa e sinais de controle enviados, via porta paralela, para a placa de acionamento da máquina CNC em utilização.

A programação CNC compreende a preparação dos dados para usinagem da peça pela máquina, através do arquivo do programa da peça em G-Code com informações contidas no programa que são referentes às dimensões e qualidades da peça, parâmetros de usinagem, funções de máquina, dados de ferramentas e informações tecnológicas, que permitam que a máquina produza a peça de forma automática.

6.1 Funções Gcode

A estrutura de um programa CNC baseia-se em funções composta de:

f) cabeçalho de programa: ligação dos eixos e configuração da área de usinagem, espessura do material e medidas da ferramenta de corte;

g) comentários: informação não executáveis, detalhes específicos para auxiliar o usuário;

h) instrução de comando: funções ‘G’ utilizadas no programa;

i) blocos de usinagem: contém informações da trajetória de usinagem;

j) fim de programa: reposiciona os eixos em ponto inicial ou posição zero máquina.

No bloco de usinagem se realiza o programa da trajetória de todos os eixos por etapas, a cada linha gera um movimento sobre um determinado eixo, seu endereço de posição é implementado por um valor numérico e a cada posição preparatória é seguida por uma determinada função ‘G’, no apêndice C representa os códigos e funções padrão utilizados no estudo deste projeto, que define o modo de movimentação, o tipo de interpolação e o sistema de medidas da máquina.

O movimento de posicionamento sempre ocorre da posição na qual se aproximou em último lugar para a posição de destino programada, esta posição de destino é por sua vez a posição de partida para o próximo comando de deslocamento, e assim sucessivamente, a instrução de programação implementada neste protótipo se baseia nas coordenadas absolutas (G90) e para trajetória de ferramenta sobre as coordenadas absolutas foi utilizado:

d) G00: interpolação linear com avanço rápido;

e) G01: interpolação linear com avanço programado;

f) G02 ou G03: interpolação circular com avanço programado.

6.2 Algoritmo de Bresenham para interpolação circular

Visando atender a proposta deste trabalho foi implementado no código padrão G o algoritmo de Bresenham para representar trajetórias de segmento em circunferências possíveis de programar.

A geração de trajetórias circulares utilizando motores de passo é tarefa mais complexa do que interpolar retas, além de exigir dos motores uma taxa de rotação constantemente variável, a derivada varia senoidalmente ao longo de um círculo, há geralmente grande esforço computacional no cálculo das equações que definem os pontos da trajetória, as quais envolvem multiplicações em ponto flutuante e extrações de raízes quadradas.

O estudo da programação com algoritmo de Bresenham evidenciou, basicamente, três tipos principais de abordagens para interpolações circulares:

d) a determinação dos pares ordenados de coordenadas cartesianas (X, Y) a partir da equação que define uma circunferência, $X^2+Y^2 = R^2$, onde R é o raio da circunferência;

e) divisão da circunferência em uma série de segmentos de reta;

f) algoritmos incrementais para o cálculo dos pontos da trajetória.

Com o algoritmo de Bresenham é capaz de calcular todos os pontos de uma circunferência usando apenas somas, subtrações e operações de multiplicar por dois, embora, teoricamente, este tipo de algoritmo seja o mais adequado do ponto de esforço computacional.

6.2.1 Interpolação circular G02 e G03

As funções G02 e G03 realiza movimentos de interpolação circular com a ferramenta sobre os sentidos dos eixos X e Y em avanço programado.

G02 – Sentido horário.

G03 – Sentido anti-horário.

Para testar o algoritmo de Bresenham foi utilizado a função G02 para interpolação circular horária da ferramenta em eixo Z sobre os eixos X e Y e criado um desenho gráfico de um raio ou semicírculo medindo 25 mm com pontos de curvatura linear enumerados. conforme figura 15.

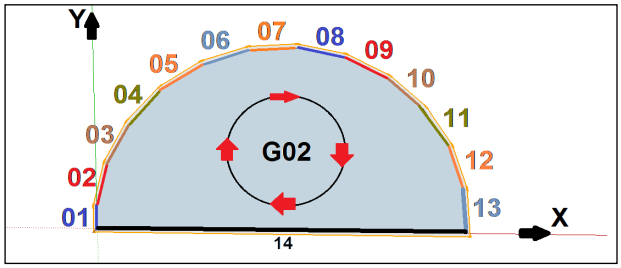


Figura 15. Interpolação G02

Cada curvatura enumerada equivale a uma linha de programa de usinagem dentro das coordenadas, para que aconteça a trajetória se deve colocar o ponto inicial do eixo X subsequente do ponto inicial do eixo Y e o ângulo referente a circunferência a ser usinada, na figura 16 apresenta as linhas do programa com o raio implementado como R25 em todas as funções até finalizar o comando.

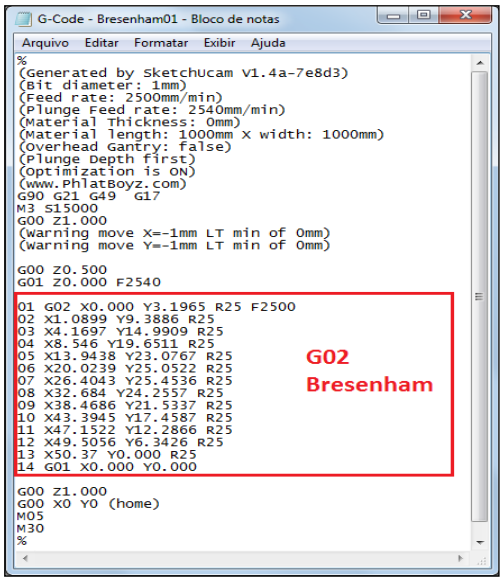


Figura 16. Algoritmo de Bresenham

7. Referência

BARBOZA, F. J. R.; AZEVEDO, J.; OLVEIRA, L. R; ALBUQUERQUE, M. L. **Uma Metodologia para Construção de Robôs Móveis.** Programa de Pós-Graduação em

Mecatrônica, Departamento de Ciência da Computação Departamento de Engenharia Mecânica, Universidade Federal da Bahia, I Seminário Técnico-Científico, I STEC, Salvador, 2003, 7p. Disponível em: <http://www.lucianooliveira.eti.br/publicacoes_files/Ogumbot.pdf>. Acesso em: 24 abr. 2007.

BRAGA, C. N. **Os Segredos da Porta Paralela.** Revista Mecatrônica Atual, São Paulo-SP, ano 1, n.º 1, nov/2001, 32p.

BRAGA, N. C. **Eletrônica Básica para Mecatrônica.** 1.ed. São Paulo: Editora Saber, 2005, 160p.

CONRADO, R. Grbl v0.9j O que é? Para que serve? Como configurar? **Atividade Maker**, [S.l], v. online, 2018. Disponível em:<http://atividademaker.com.br/grbl- v09j>Acesso em: 20 mar.2018.

COSTA, D. D.; PEREIRA, A. G. **Desenvolvimento e Avaliação de uma Tecnologia de Baixo Custo para Programação CNC em Pequenas Empresas.** Revista Produção, Abr 2006, Vol.16, Nº1, p.48-63. Universidade Federal do Paraná, 16p. Disponível em: http://www.scielo.br/

COSTA, E. R. F. **Simulação de Movimento e Planejamento de Trajetórias para Robôs Manipuladores.** Dissertação de Mestrado, Universidade Federal de Pernambuco, Recife-PE, 2003, 74p.

COSTA, E. S; SANTOS, D. J. **Processos de Usinagem.** Centro Federal de Educação Tecnológica de Minas Gerais, Curso Técnico em Eletromecânica. Divinópolis, 2006.

FEDEL, R. **Metodologia de Desenvolvimento de Equipamento para Alimentar Deficientes Físicos.** Dissertação de Mestrado da Universidade Estadual Paulista Júlio Mesquita/Ilha Solteira, Curso de Engenharia Elétrica, São Paulo-SP, 2004, 172p.

FERREIRA, J.; MARTINS, N.; AGNELO, L.; DIAS, J. **Imagiologia Tridimensional Digital para Construção de Protótipos Industriais.** Coimbra- Portugal, Instituto de Sistemas e Robótica, Universidade de Coimbra, 2001, 11p.

FILHO, J. B. M. **Controlador Adaptativo Neural para Mesa de Coordenadas X-Y.**

Curso de Pós-Graduação em Engenharia Mecânica, CT- UFPB, 2007.

FORTIN, E. **An Innovative Software Architecture to Improve Information flow from CAM to CNC.** Computers & Industrial Engineering, Vol. 46, 2004, p. 655-667.

GOELLNER, E. **Ferramenta Computacional para Acionamento de Motores de Passo Aplicados ao Projeto de Equipamentos CNC.** Dissertação de Mestrado, Santa Maria-RS, 2006, 130p.

ISO. **International Organization for Standardization.** Disponível em:

http://www.iso.org. Acesso em: 31 mai. 2008.

KAIKKONEN, J.; MAKELAINEN, T.; HAKALA, H. **Advanced Design Methods and**

Tools for Mobile Robot Development, Intelligenc for Mechanical Systems.

Proceedings IROS, 1991, 7p.

KELLYWARE. **Manual KCAM 4 - CNC CONTROL SOFTWARE**. 2007, 31p.

KORTENKAMP, D.; BONASSO, R. P.; MURPHY, R. **Artificial Intelligence and Mobile Robots: case studies of successful robot systems**, The MIT Press,

1998.

LOPES, L. C. G. **Programando para Controle de Dispositivos pelo Computador.** Informática Industrial/Automação, SDM - Sistemas Digitais e Microprocessados CEFET-MG Campus III - Uned Leopoldina, 2007, 36p.

LYNCH, M. **Computer Numerical Control: Acessory Devices**. New York: McGraw-Hill, 1994, 262p.

LYNCH, M. **The Key Concepts of CNC.** Modern Machine Shop, Cincinnati, Vol. 69, Nº 11A, 1997, p.81-144.

MACH3. **Manual de Utilização do Software Mach3.** 2005, 43p.

MACHADO, A. **Comando Numérico Aplicado às Máquinas-Ferramenta.** 4ª ed., São Paulo, Editora Ícone, 1990, 312 p.

NBR. **Associação Brasileira de Normas Técnicas.** Disponível em:

http://www.abnt.org.br. Acesso em: 31 mai. 2008.

PAZOS, F. A; LOVISOLO, L. **Automação de Sistemas e Robótica.** 1 ed. Rio de Janeiro: Editora Axcel Books do Brasil, 2002, 384p.

PENTEADO, F. A. C. A. **Processo de Usinagem dos Metais.** Revista CADware(r) Publishing Brazil. Agosto/2002.

PERACETTA, L. F.; UNIANDRADE, A. R. **Sistema de Aquisição de Dados e Controle de Processos Através da Porta Paralela.** Simpósio, 2003, 10p. Disponível em: http://www.uniandrade.br/simposio/pdf/comp105.pdf. Acesso em: 23 set. 2008.

PEREIRA, A. G. **Desenvolvimento e Avaliação de um Editor para Programação CN em Centros de Usinagem.** Dissertação de Mestrado, Curitiba-PR, 2003, 122p.

ANEXO(S)

ANEXO A – Apostila de configuração Firmware GRBL (CONRRADO, 2018)

grbl v0.9j

O que é? Para que serve? Como configurar?

$0=10 (step pulse, usec)
$1=25 (step idle delay, msec)
$2=0 (step port invert mask:00000000)
$3=6 (dir port invert mask:00000110)
$4=0 (step enable invert, bool)
$5=0 (limit pins invert, bool)
$6=0 (probe pin invert, bool)
$10=3 (status report mask:00000011)
$11=0.020 (junction deviation, mm)
$12=0.002 (arc tolerance, mm)
$13=0 (report inches, bool)
$20=0 (soft limits, bool)
$21=0 (hard limits, bool)
$22=0 (homing cycle, bool)
$23=1 (homing dir invert mask:00000001)
$24=50.000 (homing feed, mm/min)
$25=635.000 (homing seek, mm/min)
$26=250 (homing debounce, msec)
$27=1.000 (homing pull-off, mm)
$100=314.961 (x, step/mm)
$101=314.961 (y, step/mm)

ATIVIDADE MAKER

RODRIGO CONRADO

Quem somos

Atividade Maker é um projeto com o propósito de divulgar o estilo de vida "faça você mesmo" (DIY – Do It Yourself), aqui vamos divulgar projetos de Marcenaria, Eletrônica, Robótica, Arduino, Impressão 3D, CNC entre outros, Somos apaixonados pelo estilo Faça Você Mesmo.
Atividade Maker - Construa o Seu Mundo, Você Pode!

Como entrar em contato?

Você pode entrar em contato pelos seguintes meios:

www.atividademaker.com.br

www.facebook.com/atividademaker

www.youtube.com/c/AtividadeMakerOficial

contato@atividademaker.com.br

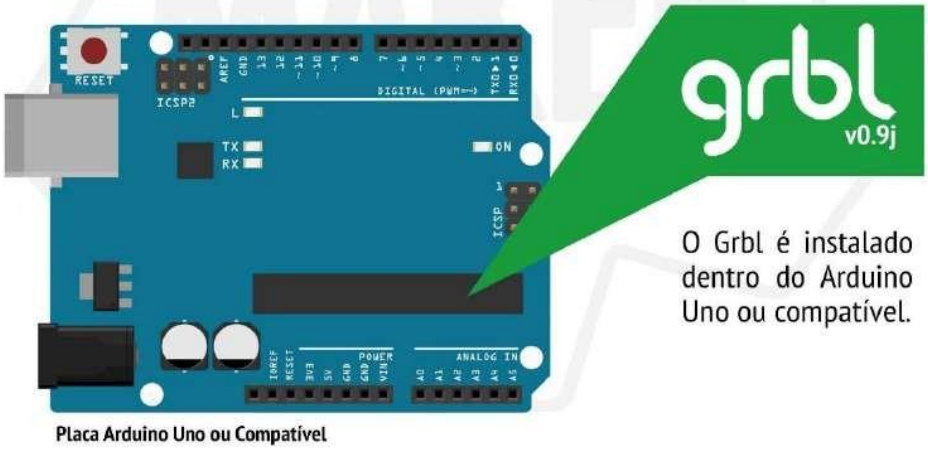
grbl v0.9j

Sobre o Grbl

O Grbl é grátis, open source (código aberto), software de alto desempenho para controlar o movimento de máquinas que se movem, que fazem as coisas, ou que fazem as coisas se moverem, e será executado diretamente em um Arduino.

Muitas impressoras 3D de código aberto usam Grbl em seus núcleos, sendo adaptado para uso em centenas de projetos, incluindo máquinas de corte a laser, CNC entre outros. Devido ao seu desempenho, simplicidade e requisitos de hardware, o Grbl tornou-se um fenômeno de código aberto.

Em 2009, Simen Svale Skogsrud (http://bengler.no/grbl) agraciou a comunidade de código aberto, escrevendo e liberando as primeiras versões do Grbl a todos (inspiradas no Arduino Gcode Interpreter por Mike Ellery). Desde 2011, Grbl está a avançar como um projeto de código aberto voltada para a comunidade sob a liderança pragmática da Sungeun K. Jeon Ph.D. (@chamnit).



Placa Arduino Uno ou Compatível

ATIVIDADE MAKER

grbl v0.9j

Primeiros Passos - Arduino

Vamos instalar o Grbl no Arduino através da IDE do Arduino, para isso você precisa acessar o site **www.arduino.cc** e efetuar o download da IDE para o seu sistema operacional.

1 - Acessar www.arduino.cc;
2 - Clicar em Download;
3 - Clicar no seu sistema operacional para efetuar o download;
4 - Faça a instalação conforme seu sistema;

5 - Abra a IDE Arduino;
6 - Clique em **Ferramentas > Porta e selecione a porta serial.**

7 - Clique em **Ferramentas > Placa e selecione Arduino Uno;**

ATIVIDADE MAKER

grbl v0.9j

Primeiros Passos - Grbl

Após a instalação da IDE Arduino e suas configurações você vai precisar efetuar o download do Grbl para que possa ser enviado para a memória do Arduino.

1 - Acesse: **https://github.com/grbl/grbl** e faça o download do Grbl clicando no botão **Download ZIP**;

2 - Descompactar o arquivo em um local de sua preferencia;

3. - Abra o IDE Arduino;

4 - Clique em **Sketch > Include Library > add .Zip Library** e selecione a pasta que você descompactou o Grbl;

5 - Conecte o cabo USB no seu computador e na sua placa Arduino;

6 - Clique em **Arquivos > Exemplos > e escolha o grbl > grblUpload**;

7 - Depois clique para transferir o código para o Arduino.

grbl v0.9j

Primeiros Passos - Terminal Serial

Pronto agora você ja tem o Grbl instalado dentro do Arduino, você ja pode fechar a IDE do Arduino pois não iremos mais utilizá-la.

Agora é preciso escolher um **Terminal Serial** para que possamos utilizar o Grbl.

Você encontra uma lista com várias opções no seguinte endereço:
https://github.com/grbl/grbl/wiki/Using-Grbl

Iremos utilizar em nosso tutorial o terminal serial, **Universal Gcode Sender** na versão **1.0.9.**

Você pode efetuar o download no seguinte endereço;
https://github.com/winder/Universal-G-Code-Sender

1 - Conecte o cabo USB no seu computador e em seu Arduino Uno;

2 - Clique em **Port** e selecione a porta serial;

3 - Clique em **Baud** e selecione 115200;

4 - Clique em Open;

5 - Se tudo ocorreu bem você deve ver a seguinte mensagem:

Grbl 0.9j ['$' for help]

ATIVIDADE MAKER

grbl v0.9j

Iniciando o Grbl v0.9j

Primeiro, conecte-se ao Grbl usando o terminal serial de sua escolha, em seguinda selecione a porta serial onde seu arduino está instalado e defina a taxa de transmissão (Baud) para 115200, clique em Open para se conectar.

Uma vez conectado você deverá visualizar o aviso, que se parece com isso:

Grbl 0.9j ['$' for help]

Configurando o Grbl v0.9j

Para visualizar as configurações, digite na linha de comando **$$** e pressione Enter.

O Grbl deve responder com uma lista das configurações atuais do sistema.

Todas essas configurações são persistentes e mantidas na memória EEPROM, então se você desligar seu Arduino os dados serão mantidos na memória e na próxima vez que você ligar eles estarão do mesmo jeito que você deixou.

grbl
v0.9j

Configurações do Sistema

Você pode alterar qualquer valor de configuração digitando na linha de comando o **$** mais o número do parâmetro que você quer alterar e informar o valor a ser inserido.

$X = valor (Salva o valor no Grbl)

O $x=valor salva ou altera um valor de configuração no Grbl, o que pode ser feito manualmente, enviando este comando quando conectado ao Grbl através de um programa terminal serial.

Para alterar manualmente, por exemplo, a opção $0 para 10us você deve digitar:

$0 = 10 (depois dar um enter)

Se tudo correu bem, o Grbl vai responder com um **'ok'** e essa configuração é armazenada na memória EEPROM e ficará lá para sempre ou até que seja alterada novamente. Você pode verificar se o Grbl recebeu e armazenou sua nova configuração corretamente, digitando **$$** para ver as configurações do sistema novamente.

```
$0=10 (step pulse, usec)
$1=25 (step idle delay, msec)
$2=0 (step port invert mask:00000000)
$3=6 (dir port invert mask:00000110)
$4=0 (step enable invert, bool)
$5=0 (limit pins invert, bool)
$6=0 (probe pin invert, bool)
$10=3 (status report mask:00000011)
$11=0.020 (junction deviation, mm)
$12=0.002 (arc tolerance, mm)
$13=0 (report inches, bool)
$20=0 (soft limits, bool)
$21=0 (hard limits, bool)
$22=0 (homing cycle, bool)
$23=1 (homing dir invert mask:00000001)
$24=50.000 (homing feed, mm/min)
$25=635.000 (homing seek, mm/min)
$26=250 (homing debounce, msec)
$27=1.000 (homing pull-off, mm)
$100=314.961 (x, step/mm)
$101=314.961 (y, step/mm)
$102=314.961 (z, step/mm)
$110=635.000 (x max rate, mm/min)
$111=635.000 (y max rate, mm/min)
$112=635.000 (z max rate, mm/min)
$120=50.000 (x accel, mm/sec^2)
$121=50.000 (y accel, mm/sec^2)
$122=50.000 (z accel, mm/sec^2)
$130=225.000 (x max travel, mm)
$131=125.000 (y max travel, mm)
$132=170.000 (z max travel, mm)
```

ATIVIDADE MAKER

grbl v0.9j

Parâmetros do Grbl v0.9j e seus significados

NOTA: A numeração de Configurações mudou desde a v0.8c.

$0 = Tamanho do Pulso em microsegundos

Os drivers controladores de motor são projetados para entender o passo com um determinado tamanho de pulso. Verifique o data sheet ou apenas faça testes com alguns valores. Você precisa de pulsos curtos mas que sejam reconhecidos pelo seu driver. Se os pulsos forem muito longos você pode ter problemas ao executar o sistema em altas taxas de alimentação de pulsos, pois os pulsos de passos podem começar a se sobreporem uns aos outros. Recomendamos algo em torno de 10 microsegundos, que é o valor padrão.

$1 = Atraso da Inatividade do Motor em milisegundos

Toda vez que seus motores completam um movimento e chegam a uma parada, o Grbl vai atrasar o desativamento dos motores por este valor. Ou você pode sempre manter seus eixos habilitados (alimentados de forma a manter a posição), definindo esse valor para o máximo de 255 milisegundos. Mais uma vez apenas para fixar você pode manter todos os eixos sempre habilitados pela configuração $1=255.

O atraso da inatividade do motor é o tempo de duração que o Grbl irá manter os motores travados antes de desativar. Dependendo do sistema, você pode definir isso para zero e desativá-lo. Em outros, você pode precisar de 25 - 50 milisegundos para garantir que seu eixo chegue a uma parada completa antes de desativá-lo.

grbl v0.9j

$2 = Máscara para Inversão do Sinal dos Passos

Este item faz a inversão do sinal de pulso. Por padrão, um sinal começa de nível baixo e vai a alto, depois de um tempo definido em **$0**, o valor volta para baixo, até o próximo evento do pulso. Quando invertido, o comportamento do pulso muda de alto para baixo depois de um tempo definido em **$0** volta para alto.

A maioria dos usuários não precisará usar essa configuração, mas isso pode ser útil para determinados controladores CNC que têm exigências peculiares.

Esta máscara para inversão dos passos é um valor que armazena os eixos para serem invertidos com bit. Você realmente não precisa entender completamente como funciona. Você só precisa digitar o valor de configurações para os eixos que deseja inverter. Por exemplo, se você quiser inverter os eixos X e Z, você irá enviar $2=5.

Valor	Máscara	Inverter X	Inverter Y	Inverter Z
0	00000000	Não	Não	Não
1	00000001	Sim	Não	Não
2	00000010	Não	Sim	Não
3	00000011	Sim	Sim	Não
4	00000100	Não	Não	Sim
5	00000101	Sim	Não	Sim
6	00000110	Não	Sim	Sim
7	00000111	Sim	Sim	Sim

grbl v0.9j

$3 = Máscara para Inversão da Direção

Esta configuração inverte o sinal de sentido para cada eixo. Por padrão, o Grbl assume que os eixos vão se mover em uma direção positiva quando o sinal de sentido é baixo, e uma direção negativa quando o sinal é alto. Muitas vezes, os eixos não se movem desta forma em algumas máquinas.

Esta definição irá inverter o sinal de sentido dos eixos.

Esta máscara para inversão da direção funciona exatamente como a máscara para inversão dos passos. Para configurar essa opção, você só precisa enviar o valor para os eixos que deseja inverter. Use a tabela anterior. Por exemplo, se quiser inverter apenas a direção do eixo Y, você envia $3=2

$4 = Inverter o Enable do Controlador

Por padrão, o enable desativa os motores com nível alto e ativa com nível baixo.

Se você precisar inverter essa condição apenas digite $4=1 se quiser deixar como padrão deixe $4=0 (Pode ser necessário desligar a alimentação do arduino e ligar novamente para carregar as mudanças).

grbl v0.9j

$5 = Inverter os Limites

Por padrão, os limites estão normalmente em alta com resistência de pull-up interno do Arduino. Quando um limite é baixo, o Grbl interpreta isso como disparado. Para o comportamento oposto, basta inverter os limites digitando $5=1. Desativar com $5=0. (Pode ser necessário desligar a alimentação do arduino e ligar novamente para carregar as mudanças).

NOTA: Se inverter seu limite, você precisará de um resistor pull-down externo para todos os limites para evitar a sobrecarga e fritá-los.

$6 = Inverter o Probe

Por padrão, o probe é realizado normalmente em alta com resistência de pull-up interno do Arduino. Quando o probe é baixo, o Grbl interpreta isso como disparado. Para o comportamento oposto, basta inverter o probe pela digitação de $6=1. Desativar com $6=0. (Pode ser necessário desligar a alimentação do arduino e ligar novamente para carregar as mudanças).

NOTA: Se você inverter o probe, você precisará de um resistor pull-down externo com o probe para evitar a sobrecarga e fritá-lo.

grbl
v0.9j

$10 = Relatório de Status

Esta configuração determina que o Grbl retorne em tempo real um relatório de status. Por padrão o Grbl irá enviar de volta o seu estado de execução (não pode ser desligado), a posição da máquina, e do trabalho (posição da máquina com coordenadas as compensações e outras compensações aplicadas). Três recursos de relatórios adicionais estão disponíveis, que são úteis para interfaces ou usuários que criam suas máquinas, que incluem o buffer serial RX, o uso do bloco do buffer, e os estados limites (como alto ou baixo, mostrado na ordem ZYX).

Para configurá-los, use a tabela abaixo para determinar quais os dados que você gostaria que o Grbl enviasse de volta. Some os valores dos tipos que você gostaria de ver nos relatórios de status. Este é o valor que você usa para enviar para o Grbl. Por exemplo, se você precisa de posição de máquina e de trabalho, some os valores 1 e 2 e envie ao Grbl $10=3. Ou, se você precisa da posição da máquina somente e estado dos limites, some os valores 1 e 16 e envie ao Grbl $10=17.

Em geral, é melhor utilizar essa configuração somente se necessário pois eleva a utilização de recursos desnecessariamente.

Tipo de Relatório	Valor
Posição da Máquina	1
Posição de Trabalho	2
Buffer	4
RX Buffer	8
Limites	16

ATIVIDADE MAKER

grbl v0.9j

$11 = Desvio de Junções

O desvio de junção é usado pelo sistema que controla a aceleração para determinar quão rápido ele pode mover-se através de junções de segmentos gerados pelo programa g-code. Por exemplo, se o seu programa g-code tem uma curva acentuada de 10 graus chegando e a máquina está se movendo na velocidade máxima, está definição ajuda a determinar o quanto a máquina precisa desacelerar para ir com segurança sem perder passos.

Calcular isso é um pouco complicado, mas em geral, valores mais elevados dão movimentos mais rápidos, enquanto aumenta o risco de perder passos e posicionamento. Os valores mais baixos fazem com que o sistema que controla a aceleração seja mais cuidadoso levando a máquina a se mover mais lentamente. Então se você quer que sua máquina faça as curvas mais rapidamente você deve aumentar esse valor para acelerar o movimento.

$12 = Tolerância de Arco em milímetros

O Grbl faz círculos G2 e G3, subdividindo-os em linhas menores, tal que a precisão do traçado nunca seja inferior a este valor. Você provavelmente nunca terá de ajustar esta definição, uma vez que 0,002 milímetros está bem abaixo da precisão da maioria das máquinas CNC.

Mas se você achar que seus círculos são realizados lentamente, ajuste essa configuração. Os valores mais baixos vão dar maior precisão, mas pode levar a problemas de desempenho, os valores mais elevados levam a uma precisão inferior, mas pode acelerar o desempenho.

grbl v0.9j

$13 = Relatório em Polegadas

O Grbl tem um recurso de relatório de posicionamento em tempo real para fornecer um feedback ao usuário de onde a máquina está exatamente nesse momento, bem como, os parâmetros para coordenar as compensações e probe. Por padrão, ele é definido para relatar em mm, mas enviando um comando $13=1, o relatório será exibido em polegadas. $13=0 para voltar para mm.

$20 = Limites Através do Software

Os limites através do software são um recurso de segurança para ajudar a prevenir que sua máquina tente se deslocar além dos limites disponíveis de curso, fazendo com que bata ou quebre alguma coisa. Ele funciona por saber os limites máximos de cada eixo e sabendo a posição atual de trabalho da máquina, quando um novo código g-code é enviado para o Grbl, ele vai verificar se é possível efetuar o movimento sem causar acidentes excedendo os limites da máquina. Configurando os limites de software o Grbl emitirá um bloqueio imediato de avanço desligando todos os movimentos e exibindo um alarme indicando o problema.

NOTA: Limites através do software requer que homing seja habilitado e as configurações máximas de cada eixo também, porque o Grbl precisa saber onde ele está. $20=1 para habilitar, e $20=0 para desabilitar.

grbl v0.9j

$21 = Limites de Hardware

Limites de hardware tem basicamente a mesma função que limites de software, mas usam chaves fim de curso para controlar de forma física os movimentos. Basicamente você conecta algumas chaves (mecânicas, magnéticas ou ópticas) perto do final do curso de cada um dos eixos, ou onde quer que você ache necessário que o movimento seja interrompido. Quando o interruptor dispara, ele irá parar imediatamente todo o movimento e entrar em modo de alarme, o que o obriga a você verificar a sua máquina.

Para utilizar os limites de hardware com o Grbl, os limites são mantidos em nível alto com um resistor pull-up interno, então tudo que você tem a fazer é colocar um interruptor normalmente aberto com o terra e definir limite de hardware com $21=1. (Desativar com $21=0) Aconselhamos a tomar medidas de prevenção de interferência elétrica. Se você quiser um limite para ambas as extremidades do curso do eixo, ligue os fios em paralelo assim qualquer um deles irá acionar o limite.

Tenha em mente que um evento de limite de hardware é considerado um evento crítico, onde os motores param imediatamente. O Grbl não terá nenhum feedback sobre onde e qual o posicionamento da sua máquina e vai entrar em um modo de loop infinito ALARME, dando-lhe a oportunidade de verificar sua máquina e forçá-lo a redefinir o Grbl. Lembre-se que é puramente um recurso de segurança.

grbl v0.9j

$22 = Ciclo de Homing

Para aqueles que estão iniciados no mundo do CNC, o ciclo de homing é usado para localizar uma posição conhecida e consistente em uma máquina cada vez que você iniciar o seu Grbl entre as sessões. Em outras palavras, você sabe exatamente onde você está em um determinado momento, de cada vez. Digamos que você começa a usinagem de algo ou está prestes a iniciar a próxima etapa em um trabalho e você precisa parar, quando reiniciar o Grbl ele não terá idéia de onde ele estava. Se você tem homing, você tem sempre o ponto de referência zero da máquina, a partir dai o que você tem a fazer é executar o ciclo de homing e retomar de onde parou.

Para configurar o ciclo de homing para o Grbl, você precisa ter interruptores de limite em uma posição fixa, ou então o seu ponto de referência fica confuso. Normalmente, eles são configurados no ponto mais distante em X+, Y+, e Z+ de cada um dos eixos. Por padrão, no ciclo de homing o Grbl move o eixo Z positivo primeiro para limpar a área de trabalho e, em seguida, move os eixos X e Y ao mesmo tempo no sentido positivo.

Quando o homing está habilitado o Grbl vai bloquear todos os comandos g-code até que você execute um ciclo de homing. Nenhum eixos se movimentará, a menos que o bloqueio seja desativado com ($X).

$22=1 habilita o homing e $22=0 volta a desativar.

$x = desabilita mensagem de alerta e $h = Procura o homing

NOTA: Confira o config.h para mais opções de homing para usuários avançados. Você pode desativar o bloqueio de homing na inicialização, configurar quais os eixos que se movem pela primeira vez durante um ciclo de homing e em que ordem, e muito mais.

ATIVIDADE MAKER

grbl v0.9j

$23 = Máscara para Inverter a Direção do Homing

Por padrão, o Grbl assume que os seus interruptores de limite estão no sentido positivo, o primeiro movimento será no eixo Z positivo, então os próximos movimentos serão simultaneamente nos eixos X e Y positivos até localizar seus limites de curso. Se a sua máquina tem um interruptor de limite na direção negativa, a máscara para inverter a direção do homing pode inverter a direção dos eixos. Ele funciona exatamente como a inversão do passo e da direção, onde tudo que você tem a fazer é enviar o valor correspondente da tabela para indicar quais os eixos que deseja inverter e procurar na direção oposta.

$23 = Usar a tabela de mascaras

$24 = Homing Feed em mm/min

Ao iniciar o ciclo de homing sua máquina irá procurar o interruptor de fim de curso em uma velocidade mais alta, depois de encontrá-los, move-se a uma velocidade mais baixa até chegar ao ponto zero da máquina. Defina essa opção para um valor que seja possível voltar ao ponto zero sempre com precisão e repetibilidade.

$25 = Homing Seek em mm/min

É a velocidade em que sua máquina irá procurar as chaves de fim de curso. Ajuste para a maior velocidade possível mas que garanta que a máquina não colida muito fortemente com os interruptores.

grbl v0.9j

$26 = Homing Debounce em milisegundos

Sempre que um interruptor de fim de curso for acionado é necessário verificar se o acionamento foi de fato pela máquina ter encontrado o fim de curso ou se o acionamento foi dado através de ruído elétrico ou mecânico, ou seja, bounce é o sinal alto e baixo em milisegundos antes que ele se estabilize.

Para resolver isso você precisa estabilizar o sinal que poder ser por hardware com algum tipo de condicionador de sinal ou por software com um pequeno atraso da chegada do sinal.

Defina este valor de atraso para que o seu sistema possa obter homing repetível. Na maioria dos casos, de 5-25 milisegundos é bom.

$27 = Homing Pull-Off em milímetros

Quando sua máquina completar o homing ela poderá voltar um determinado valor em milímetros para impedir que o acionamento dos fim de cursos sejam acionados acidentalmente caso eles estejam configurados em ambos os lados do seu eixo.

grbl
v0.9j

$100, $101 e $102 = X,Y e Z Passos por Milímetros

O Grbl precisa saber quantos passos são necessários para mover sua ferramenta. Para calcular os passos/mm para um eixo da sua máquina você precisa saber:

- Quantos **mm** a sua máquina se moveu após uma revolução.
- Quantos passos para seu motor fazer uma revolução (normalmente 200)
- Quantos micropassos você configurou em seu controlador (tipicamente 1, 2, 4, 8 ou 16).

Dica: Utilizar valores altos de micropassos (por exemplo, 16) pode reduzir o torque do motor de passo, então use o menor que lhe dá a resolução do eixo desejado e as propriedades de funcionamento confortáveis.

Os passos/mm pode ser calculados da seguinte forma:

$$passos\_por\_mm = \frac{(passos\_por\_revolução * micropassos)}{mm\_percorridos\_revolução}$$

Calcule este valor para cada eixo e escreva essas configurações para o Grbl.

grbl
v0.9j

$110, $111 e $112 = X,Y e Z Taxa Máxima de mm/min

Isso define a taxa máxima que cada eixo pode se mover. Sempre que o Grbl executa um movimento, ele verifica antes se o movimento a ser executado irá ultrapassar a taxa máxima. Isto significa que cada um dos eixos tem a sua própria velocidade independente, que é extremamente útil para a limitação do eixo Z tipicamente mais lentos.

A maneira mais simples para determinar esses valores é testar cada eixo um de cada vez, aumentando lentamente as configurações de taxa e movê-lo. Por exemplo, para testar o eixo X, enviar ao Grbl algo como G0 X50 com suficiente distância de deslocamento de modo a que o eixo acelera a sua velocidade máxima.

Você saberá que você bateu o limiar de taxa máxima quando seus motores pararem de funcionar. Ele vai fazer um pouco de barulho, mas não irá estragar seu motor. Introduza um ajuste de 10-20% abaixo deste valor, então você pode testar novamente até chegar a uma taxa adequada. Em seguida, repita para os outros eixos.

NOTA: Esta definição de taxa de máxima também define a velocidade do G0.

grbl
v0.9j

$120, $121 e $122 = X,Y e Z Aceleração em mm/s2

Isso define os parâmetros de aceleração dos eixos em mm/segundo ao quadrado.

Com um valor mais baixo o Grbl fará movimentos mais lentos, enquanto um valor maior produz movimentos mais rapidos. Muito parecido com o ajuste da taxa máxima, cada eixo tem o seu próprio valor de aceleração e são independentes uns dos outros. Isto significa que um movimento multi-eixo só irá acelerar levando em consideração o valor do menor eixo.

A maneira mais simples para determinar os valores para esta configuração é testar individualmente cada eixo aumentando lentamente valores até atingir o limite do motor. Em seguida, finalizar a sua configuração de aceleração com um valor de 10-20% abaixo deste valor máximo absoluto.

É altamente recomendável que você teste com alguns programas g-code pois as vezes a carga em sua máquina é diferente quando se deslocam em todos os eixos juntos.

$130, $131 e $132 = X,Y e Z Tamanho da Área Útil em mm

Isso define o curso máximo de ponta a ponta para cada eixo em milímetros. Isto só é útil se você tem Limites de software (e homing) ativado, pois isso só é usado pelo recurso de limites de software do Grbl para verificar se você excedeu seus limites da máquina com um comando de movimento.

Obrigado

Espero que esse material tenha sido útil para que você aprimore seu conhecimento e faça a sua CNC funcionar da melhor forma possível.

Caso queira mais informações assista o canal Atividade Maker no youtube, lá você vai encontrar mais informações sobre o Grbl v0.9j.

Como entrar em contato?

Você pode entrar em contato pelos seguintes meios:

www.atividademaker.com.br

www.facebook.com/atividademaker

www.youtube.com/c/AtividadeMakerOficial

contato@atividademaker.com.br