

**UNIVERSIDADE DE TAUBATÉ**

**Leonardo Sanches**

**Luís Gustavo Pereira**

**PROJETO PARA DESENVOLVIMENTO DE UMA FRESA  
CNC COMANDADA POR UMA PLACA ARDUINO**

**Taubaté – SP**

**2017**

**Leonardo Sanches**  
**Luís Gustavo Pereira**

**PROJETO PARA DESENVOLVIMENTO DE UMA FRESA  
CNC COMANDADA POR UMA PLACA ARDUINO**

Trabalho de Graduação apresentado para obtenção do Título de Bacharel em Engenharia de Controle e Automação do Departamento de Engenharia Mecânica da Universidade de Taubaté.

Orientador: Prof. Dr. Álvaro Manoel de Souza Soares

**Taubaté – SP**

**2017**

**Ficha Catalográfica elaborada pelo SIBi – Sistema Integrado  
de Bibliotecas / Unitau - Biblioteca das Engenharias**

Pereira, Luis Gustavo

P436p Projeto para desenvolvimento de uma fresa CNC comandada por uma placa Arduino. / Luis Gustavo Pereira, Leonardo Sanches. - 2017.

60f. : il; 30 cm.

Monografia (Graduação em Engenharia de Controle e Automação) – Universidade de Taubaté. Departamento de Engenharia Mecânica e Elétrica, 2017

Orientação: Prof. Dr. Álvaro Manoel de Souza Soares, Departamento de Engenharia Mecânica e Elétrica.

1. Automação. 2. CNC. 3. Circuito impresso. I. Título.

Leonardo Sanches  
Luís Gustavo Pereira

PROJETO PARA DESENVOLVIMENTO DE UMA FRESA CNC COMANDADA POR  
UMA PLACA ARDUINO

Trabalho de Graduação apresentado para  
obtenção do Título de Bacharel em  
Engenharia de Controle e Automação do  
Departamento de Engenharia Mecânica  
da Universidade de Taubaté.

Data: 21 de novembro de 2017

Resultado: Aprovado

BANCA EXAMINADORA

Prof. Dr. Alvaro Manoel de Souza Soares

Universidade de Taubaté

Assinatura



Prof. MSc. Fabio Santajani

Universidade de Taubaté

Assinatura



“Demore o tempo que for para decidir o que você quer da vida, e depois que decidir não recue ante nenhum pretexto, porque o mundo tentará te dissuadir.”- Friedrich Nietzsche

## **AGRADECIMENTOS**

Agradecemos aos nossos pais, amigos e colegas de cursos que nos deram forças para continuar em nossa jornada. Aos professores que nos orientaram nesse caminho e nos proporcionaram conhecimento e nos auxiliaram em nossa formação profissional e pessoal.

Sanches, Leonardo. Pereira, Luis. **Projeto para desenvolvimento de uma fresa CNC comandada por uma placa Arduino**. 2017. xxx f. Trabalho de Graduação (Bacharel em Engenharia de Controle e Automação) – Departamento de Engenharia Mecânica – DEM, Universidade de Taubaté, Taubaté. Brasil.

## **RESUMO**

Placas de circuito impresso se encontram em quase todo produto que envolva eletrônica, itens de informática, telecomunicações, bens de consumo, módulos de eletrônica embarcada e de automação. As placas de circuito impresso ou também conhecido como PCI, são compostas em placas isolantes onde é depositada uma trilha de cobre que fornece contato com os componentes (resistores, capacitores, transistores, diodos e etc.) do circuito. O objetivo desse trabalho é criar uma fresa CNC automatizada controlada por um microcontrolador Arduino UNO R3 para desenhar o circuito impresso em uma placa de cobre, utilizando materiais baratos e de fácil acesso em sua construção. O presente trabalho, demonstra em seus resultados que é possível a confecção de uma fresa CNC de baixo custo para a impressão de circuitos impressos, mostrando precisão e intercambiabilidade, contudo a impressão final detém ainda algumas pequenas imperfeições, porém essas podem ser facilmente corrigidas a mão. A fresa CNC desenvolvida no presente trabalho mostrou-se uma opção de baixo custo e eficiente para a obtenção de placas de circuito impresso, diminuindo o tempo de confecção e viabilizando a impressão de projetos complexos.

**Palavras-chave:** (Automação; Circuito impresso; Fresa CNC)

Sanches, Leonardo. Pereira, Luis. **Design for the development of a CNC cutter controlled by an Arduino board**. 2017. xxx f. Graduation Work (Bachelor in Control Engineering and Automation) - Department of Mechanical Engineering - DEM, University of Taubaté, Taubaté. Brazil.

### **ABSTRACT**

Printed circuit boards are found in almost every product that involves electronics, IT items, telecommunications, consumer goods, embedded electronics and automation modules. Printed circuit boards, or also known as PCI, are composed of insulation boards where a copper track is deposited which provides contact with the components (resistors, capacitors, transistors, diodes and etc.) of the circuit. The purpose of this work is to create an automated CNC milling cutter controlled by an Arduino UNO R3 microcontroller to draw the printed circuit on a copper plate using inexpensive and easily accessible materials in its construction. The present work demonstrates in its results that it is possible to make a low cost CNC mill for printed circuit printing, showing precision and interchangeability, however the final printing still has some small imperfections, however these can be easily corrected by hand . The CNC mill developed in the present work proved to be a low-cost and efficient option to obtain printed circuit boards, reducing the time of production and making the printing of complex projects possible.

**Keywords:** (Automation; Printed circuit; CNC Milling Machine)

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>13</b>
<b>1.1 OBJETIVOS</b> .....	<b>14</b>
1.1.1 OBJETIVO GERAL.....	14
1.1.2 OBJETIVOS ESPECÍFICOS .....	14
1.2 DELIMITAÇÃO DO ESTUDO.....	14
1.3 RELEVÂNCIA DO ESTUDO .....	15
1.4 ORGANIZAÇÃO DO TRABALHO .....	15
<b>2 REVISÃO DA LITERATURA</b> .....	<b>16</b>
2.1 MICROCONTROLADOR.....	16
2.1.1 ORIGEM DOS MICROCONTROLADORES.....	16
2.1.2 MICROCONTROLADORES: CONCEITOS BÁSICOS.....	16
2.1.3 MICROCONTROLADORES: UNIDADE DE MEMÓRIA.....	18
2.1.4 MICROCONTROLADORES: UNIDADE CENTRAL DE PROCESSAMENTO (CPU) .....	20
2.1.5 MICROCONTROLADORES: BARRAMENTOS .....	21
2.1.6 MICROCONTROLADORES: UNIDADE DE SAÍDA (I/O).....	22
2.1.7 MICROCONTROLADORES: COMUNICAÇÃO SERIAL.....	22
2.1.8 MICROCONTROLADORES: TIMER.....	24
2.1.9 MICROCONTROLADORES: CONVERSOR ANALÓGICO-DIGITAL.....	26
2.1.10 MICROCONTROLADORES: PROGRAMA.....	26
2.1.11 MICROCONTROLADORES: ARDUINO .....	27
2.2 COMANDO NUMÉRICO COMPUTADORIZADO .....	28
2.2.1 COMANDO NUMÉRICO COMPUTADORIZADO: HISTÓRIA.....	28
2.2.1.1 HISTÓRIA: DESENVOLVIMENTO DOS PRIMEIROS CONTROLES..	29
2.2.1.2 HISTÓRIA: DESENVOLVIMENTO DOS COMPUTADORES.....	30
2.2.1.3 HISTÓRIA: LINGUAGEM G .....	30
2.2.2 COMANDO NUMÉRICO COMPUTADORIZADO: VANTAGENS .....	31
2.2.3 COMANDO NUMÉRICO COMPUTADORIZADO: FUNDAMENTOS DO CONTROLE NÚMERO.....	31

2.2.3.1 FUNDAMENTOS DO CONTROLE NUMÉRICO: CONTROLE DOS MOVIMENTOS.....	32
2.2.3.2 FUNDAMENTOS DO CONTROLE NUMÉRICO: SISTEMAS DE COORDENADAS .....	33
<b>3 METODOLOGIA .....</b>	<b>35</b>
3.1 MATERIAIS.....	35
3.2 MONTAGEM .....	36
3.3 GERAÇÃO DO G-CODE (VETORIZAÇÃO DA IMAGEM) .....	41
3.3.1 INKSCAPE 0.48.5 .....	42
3.3.1.2 CONFIGURAÇÃO DA ÁREA DE IMPRESSÃO .....	42
3.3.1.3 CARREGAMENTO DO DESENHO.....	43
3.3.1.4 EXTRAINDO CONTORNO OU BORDAS PARA IMPRESSÃO .....	44
3.3.1.5 GERANDO OS NÓS DE COORDENADAS .....	47
3.3.1.6 GERANDO O G-CODE .....	49
<b>4 RESULTADOS.....</b>	<b>54</b>
<b>5 CONCLUSÃO .....</b>	<b>58</b>
<b>REFERÊNCIAS.....</b>	<b>59</b>

## LISTA DE FIGURAS

Figura 1 - Ilustração interna de um microcontrolador .....	17
Figura 2- Modelo básico de um microcontrolador .....	18
Figura 3 - Unidade de memória.....	19
Figura 4 - Unidade central de processamento (CPU).....	20
Figura 5 - Barramentos .....	21
Figura 6 - Unidade de saída.....	22
Figura 7 - Bloco de comunicação serial .....	24
Figura 8 - Contador .....	25
Figura 9 - Watchdog.....	25
Figura 10 - Conversor analógico digital.....	26
Figura 11 - Resumo placa Arduino.....	28
Figura 12 - A máquina de tecer de Jacquard .....	29
Figura 13 - O movimento da mesa de uma máquina convencional.....	32
Figura 14 - Sistema de acionamento de uma máquina CNC .....	33
Figura 15 - Sistema de coordenadas retangular para o plano XY.....	34
Figura 16 - Quadrantes .....	35
Figura 17 - Materiais para estrutura .....	37
Figura 18 - Estrutura Montada.....	38
Figura 19 - Esquemático do Fritzing.....	39
Figura 20 - Eixo do motor.....	40
Figura 21 - Base de fixação do servo motor e caneta .....	41
Figura 22 - Software de edição de imagem e desenho Inkscape.....	42
Figura 23 - Configuração da Área de Impressão.....	43
Figura 24 - Carregamento do Desenho.....	44
Figura 25 - Extração Contorno ou Bordas.....	45
Figura 26 - Remoção do Background.....	46
Figura 27 - Imagem Cinza sem fundo e Imagem Vermelha com fundo .....	47
Figura 28 - Vetorização da Imagem .....	48
Figura 29 - Coordenadas Geradas em X e Y .....	49

<b>(Figura 31 - Configuração de Impressão - Fonte: Printscreen do aplicativo Inkscape)</b> .....	51
<b>(Figura 32 - IDE Processing - Fonte: Printscreen do aplicativo Inkscape)</b> .....	52
<b>(Figura 33 - Interface de Controle da CNC - Fonte: Printscreen do aplicativo Inkscape)</b> .....	53
<b>(Figura 34 - Placa 5x10 Impressa pela CNC - Fonte: Elaborado pelo Autor)</b> .....	55
<b>(Figura 35 - Placa 10x10 Impressa pela CNC - Fonte: Elaborado pelo Autor)</b> ....	56
<b>(Figura 36 - Placas 5x10 e 10x10 após tratamento químico com Percloroeto de Ferro - Fonte: Elaborado pelo Autor)</b> .....	57

## 1 INTRODUÇÃO

O desenvolvimento de placas de circuito impresso (PCI) modernas teve seu início no século 20, contudo, foi durante a segunda guerra mundial, devido à necessidade de se desenvolver circuitos para mísseis antiaéreos que resistissem ao impacto do disparo, que a tecnologia de confecção de circuitos impressos atingiu um novo patamar. Esse novo modelo de circuito impresso era feito em uma superfície de cerâmica revestida com tinta metálica para condutores, carbono para os resistores e cerâmica para os capacitores, mostrando-se viável e gerando uma patente de processo. A partir de 1980 a técnica de montagem em superfície de circuitos impressos vem sendo usada com a finalidade de se eliminar os componentes sobressalentes, com o objetivo de obter circuitos cada vez menores.

Inicialmente os projetos de circuito impresso eram feitos manualmente em uma folha de Mylar (marca registrada de uma película de poliéster), seguindo o diagrama esquemático definido, onde os *pads* dos pinos eram posicionados na placa e então as ligações entre os mesmo eram executadas com fita auto-adesiva. A fim de se padronizar o processo, hoje os circuitos são feitos em placas banhadas a cobre, onde o layout desejado é obtido com tinta especifica para o processo, e então a placa é banhada em produto corrosivo de percloroeto de ferro, permanecendo somente o layout desejado.

O layout para as placas modernas são definidos por softwares que levam em conta o tamanho da placa a ser impressa, a distribuição dos *pads* de pinos, a distribuição dos componentes a serem acoplados na placa, a necessidade da mesma, etc., gerando cada vez *layouts* mais complexos, o que torna a confecção manual das PCIs inviável. Dessa forma, uma fresa CNC, se mostra relevante, onde a precisão do comando numérico computacional (CNC) proporciona a obtenção desses layouts complexos.

As fresas comumente encontradas no mercado têm alto custo, assim o presente trabalho tem como objetivo confeccionar uma fresa CNC, utilizando

microcontrolador Arduino Uno R3, e imprimir um layout em uma placa de cobre, para demonstrar sua funcionalidade, sendo uma alternativa viável para impressão de layouts complexos

## **1.1 OBJETIVOS**

### **1.1.1 OBJETIVO GERAL**

O presente trabalho tem como objetivo geral construir uma fresa CNC (Computador Numericamente Controlado) de baixo custo para a confecção de placas de circuitos impressos, utilizando placa de controle Arduino UNO R3.

### **1.1.2 OBJETIVOS ESPECÍFICOS**

O projeto tem como primeiro objetivo a construção de uma fresa CNC controlada por Arduino UNO R3 utilizando-se de materiais de baixo custo e fácil acesso. O segundo objetivo é a impressão de um circuito pré-definido numa placa de cobre usando a fresa construída.

## **1.2 DELIMITAÇÃO DO ESTUDO**

O trabalho em questão, não visa o estudo de circuitos elétricos e nem as especificidades de circuitos impressos. Também não é do escopo do presente trabalho o desenvolvimento de código de programação e nem seu estudo.

### **1.3 RELEVÂNCIA DO ESTUDO**

Devido a complexidade dos circuitos eletrônicos é indispensável precisão na impressão dos mesmos. A fresa CNC atende essa necessidade, podendo ser construída com matérias de baixo custo e de fácil acesso, alcançando a precisão desejável.

### **1.4 ORGANIZAÇÃO DO TRABALHO**

O primeiro capítulo trata da introdução do projeto em questão, tanto quanto dos objetivos e da relevância do trabalho.

O segundo capítulo trata do fundamento para a construção da fresa CNC, dando um breve histórico da placa microcontrolador Arduino UNO R3 e demonstra o funcionamento de um microcontrolador. Também é abordada a história da CNC e suas vantagens.

O terceiro capítulo trata sobre a metodologia utilizada para a confecção da fresa CNC em questão, enumerando os itens e as etapas necessárias para se construir a mesma.

O quarto capítulo trata dos resultados do projeto em questão, ilustrando-o e demonstrando a eficiência da fresa CNC, assim como os defeitos, provenientes das limitações físicas do projeto.

O quinto capítulo trata da conclusão do projeto, onde é exposto a posição dos autores do projeto em relação ao projeto como todo, assim como o resultados.

Por fim, o sexto capítulo trata das referencias bibliográficas usadas para a confecção do projeto em questão.

## 2 REVISÃO DA LITERATURA

### 2.1 MICROCONTROLADOR

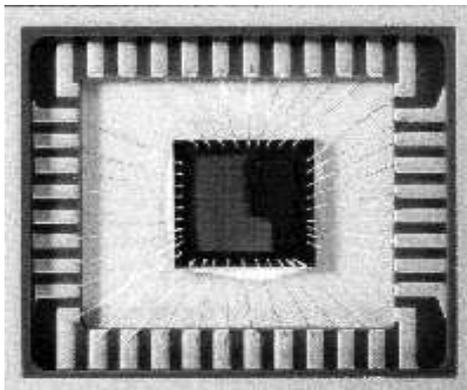
#### 2.1.1 ORIGEM DOS MICROCONTROLADORES

De acordo com Matic e Andric, foi em 1971, baseado em circuitos integrados para calculadoras de uso interno da Intel, que foi lançado o primeiro microcontrolador comercial de 4bits denominado 4004, com capacidade de 6 000 operações por segundo. Pouco tempo depois a companhia americana CTC pediu a Intel e a Texas Instruments microcontroladores de 8bits para utilização em terminais. Mesmo com a empresa CTC desistindo da ideia, em abril de 72, tanto Intel quanto Texas Instruments lançaram no mercado seus respectivos microcontroladores de 8bits, denominados 8008, com capacidade de endereçamento de memória de 16KB, 45 instruções e tinham velocidade de 300 000 operações por segundo.

#### 2.1.2 MICROCONTROLADORES: CONCEITOS BÁSICOS

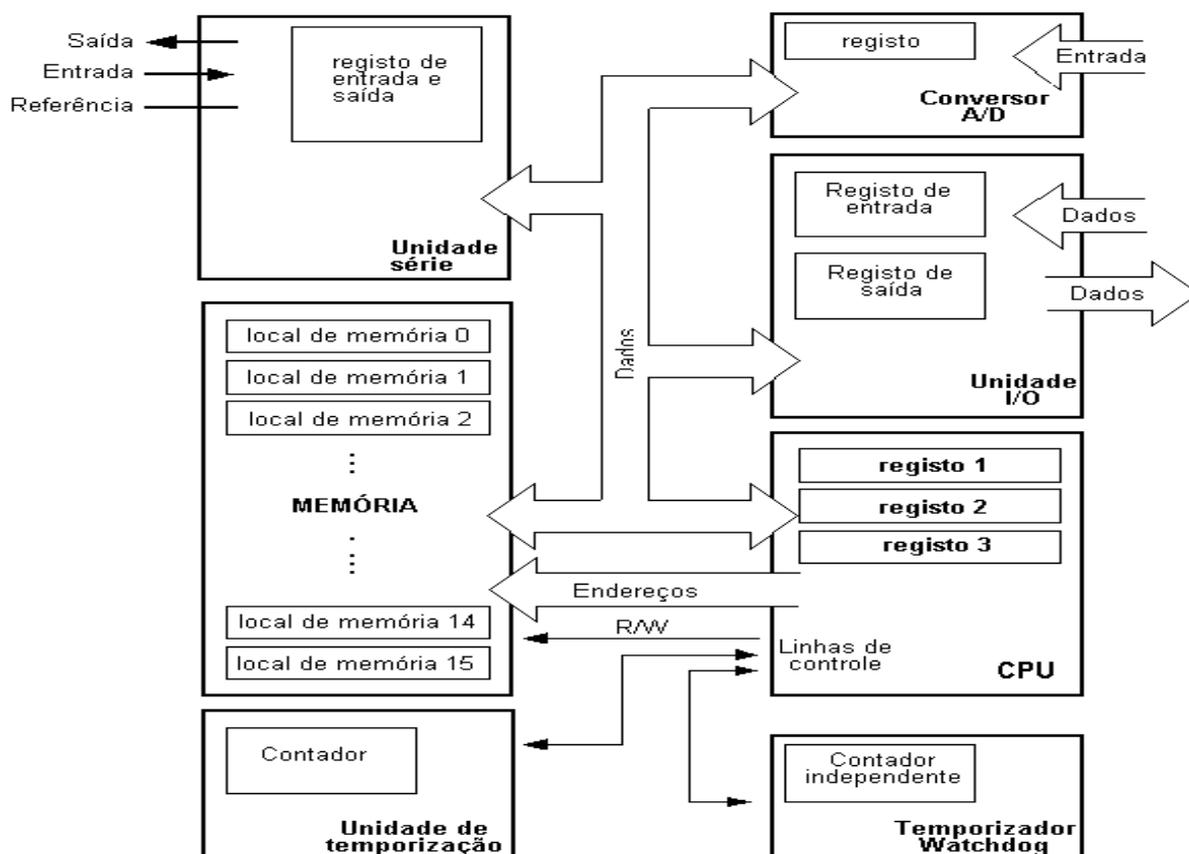
Um microcontrolador é um circuito integrado, podendo ser entendido como um sistema computacional completo. Ele é composto por alguns elementos, são eles: Memória RAM, Unidade Central de Processamento (CPU), programa ROM (para manipular, operar e armazenar dados), *Clock* para obter uma sequência do trabalho da CPU, portas binárias (I/O). É possível utilizar-se de periféricos no microcontrolador como módulos de temporização, conversores analógico/digital, conversores USB (*Universal Serial Bus*) e *ETHERNET* (Matic e Andric, 2000; Penido 2013). A Figura 1 e Figura 2 ilustram o modelo básico de um microcontrolador.

Figura 1 - Ilustração interna de um microcontrolador



Fonte: microcontroladores pic (2000)

Figura 2- Modelo básico de um microcontrolador



Fonte: microcontroladores pic (2000)

### 2.1.3 MICROCONTROLADORES: UNIDADE DE MEMÓRIA

A memória de um microcontrolador tem a função de armazenar dados, analogamente podendo ser explicada como uma prateleira com muitas gavetas. Ao atribuímos a essas gavetas valores ou símbolos de forma a não serem confundidas umas com as outras, podemos acessar esses dados armazenados facilmente, bastando saber a designação de cada gaveta. Podemos entender os componentes de uma memória exatamente assim, para cada endereço de memória temos um dado armazenado. Dessa forma, a unidade de memória é o conjunto de endereços de memória (gavetas) e endereçamento é escolher uma dessas "gavetas". É preciso selecionar o endereço de memória desejado e aguardar que o dado armazenado no

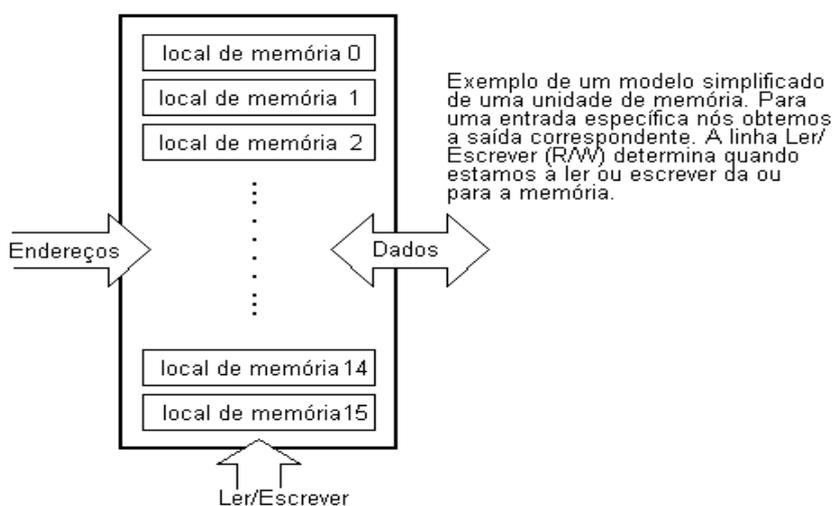
mesmo nos seja mostrado. Também é possível escrever num determinado endereço de memória, isso quer dizer, escolher um endereço e armazenar um dado. O armazenamento e/ou a leitura de um dado é feito utilizando-se de uma linha de controle que pode ser designada como R/W (Read/Write – Ler/Escrever) que é usada assim:

***r/w = 1, é executada uma operação de leitura;***

***r/w ≠ 1, é executada uma operação de escrita;***

A memória é o primeiro componente a ser considerado, contudo é preciso mais alguns para que um microcontrolador possa operar (Matic e Andric, 2000; Penido 2013). A Figura 3 ilustra o esquemático de uma unidade de memória.

Figura 3 - Unidade de memória

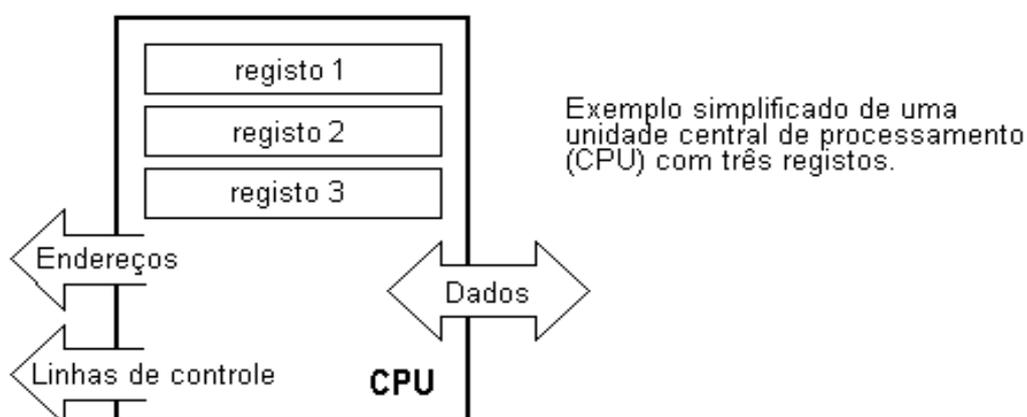


Fonte: microcontroladores PIC (2000)

### 2.1.4 MICROCONTROLADORES: UNIDADE CENTRAL DE PROCESSAMENTO (CPU)

A unidade central de processamento (CPU – central processing unit) atribui ao microcontrolador a capacidade de somar, subtrair, multiplicar e dividir, além de mover determinados conteúdos de um endereço de memória para outro. A CPU contém três locais de memória que são responsáveis pelas operações anteriormente citadas. Esses locais de memória são designados como registros e tem por finalidade o auxílio na execução de inúmeras operações, matemáticas ou não, com os dados armazenados, não importando o endereço de memória designado para os mesmos (Matic e Andric, 2000; Penido 2013), como mostra a figura 4.

Figura 4 - Unidade central de processamento (CPU)



Fonte: microcontroladores PIC (2000)

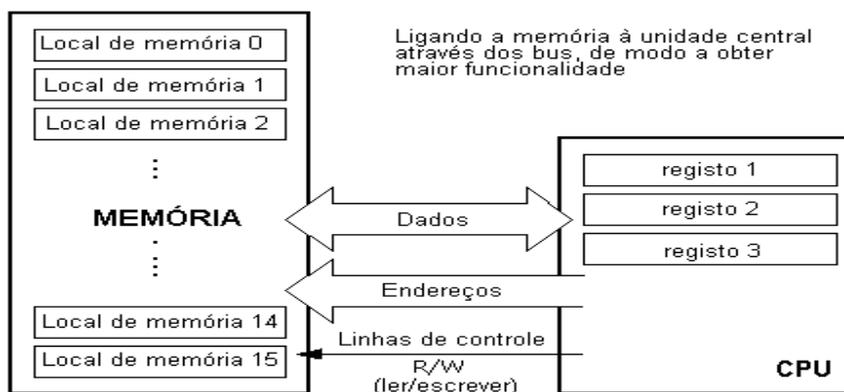
Observando a estrutura que temos até então, duas entidades independentes (CPU e memória) estão interligadas. Para que possamos acionar um conteúdo em um endereço de memória, processá-lo e novamente armazenar o resultado na memória, é preciso uma ligação entre os componentes (memória e CPU), para que esses troquem dados entre si. Essa independência entre os componentes faz com

que qualquer troca de dados entre ambos seja retardada e que a funcionalidade do sistema seja reduzida (Matic e Andric, 2000).

### 2.1.5 MICROCONTROLADORES: BARRAMENTOS

A ligação entre memória e a CPU é feita por um grupo de 8, 16 ou mais linhas que são denominados barramento e este se divide em dois tipos: barramento de dados e barramento de endereço. O número de linhas do barramento de dados depende da quantidade de memória que queremos e o número de linhas do barramento de endereço depende do tamanho do argumento de dado. Dessa forma, a aplicação do barramento de dados é na ligação da memória com a CPU e a do barramento de endereço na ligação de todos os outros elementos do microcontrolador (Matic e Andric, 2000; Penido 2013). A figura 5 a seguir ilustra um barramento.

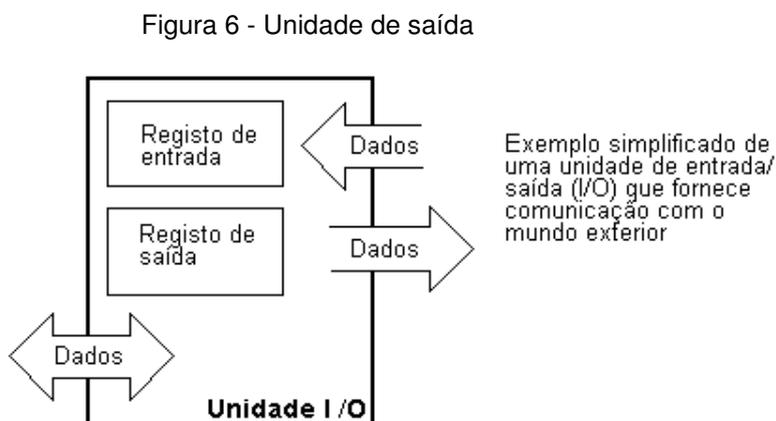
Figura 5 - Barramentos



Fonte: microcontroladores PIC (2000)

### 2.1.6 MICROCONTROLADORES: UNIDADE DE SAÍDA (I/O)

A memória e a CPU, interligadas através de barramentos, tem sua funcionalidade bem ampla, capaz de armazenar e operar dados, contudo não é possível, para nós, operarmos tais funcionalidades. A fim de atendermos essa necessidade, um novo elemento é adicionado que contém várias localizações de memória, tal elemento é ligado ao barramento de dados e a linhas de saída que coincidem com pinos do circuito integrado do microcontrolador. As recém-adicionadas localizações de memória são denominadas “portos” e estes podem ser: porto de entrada, porto de saída ou porto entrada/saída. Ao se trabalhar com esses “portos”, primeiramente é preciso determinar que tipo de porto se deseja para em seguida enviar ou receber dados para ou desse porto escolhido ((Matic e Andric, 2000; Penido 2013). A figura a seguir mostra uma unidade de saída.



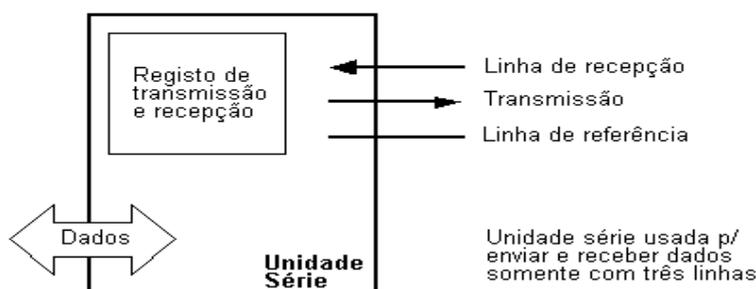
Fonte: microcontroladores pic (2000)

### 2.1.7 MICROCONTROLADORES: COMUNICAÇÃO SERIAL

A unidade de entrada e saída (I/O) nos garante uma comunicação confiável do microcontrolador com o mundo exterior, porém essa forma de comunicação tem

suas restrições. Uma das restrições mais recorrentes é o número de linhas necessárias usadas para se transmitir um dado. Dados grandes exigem um número de linhas maiores em relação dados menores para serem transmitidos. Assim, com a necessidade de se transmitir dados cada vez maiores, faz com que uma nova forma de comunicação seja utilizada, afim de se manter um número de linhas viáveis. Para entendermos essa nova forma de comunicação, imaginemos que estamos utilizando três linhas, sendo uma para enviar dados, outra para receber dados e uma linha de referência, funcionando como entrada e saída de dados. Afim que essa estrutura de três linhas funcione, é necessário criar algumas regras, denominadas protocolos, para a troca de dados. Os protocolos são previamente definidos para que não haja conflito entre o emissor e o receptor de determinado dado. Podemos exemplificar um protocolo de tal forma: um valor lógico 1 é enviado por uma linha de transmissão até que se inicie a transferência do dado em questão. Após isso, um valor lógico 0 é transmitido por um determinado tempo, dessa forma, o lado receptor “percebe” que existem dados a receber, ativando-se. Nesse momento o emissor começa a transmitir valores lógicos de 0 e 1 através da linha de transmissão, sendo a quantidade de valores lógicos enviados correspondentes ao número de bits do microcontrolador em questão, seguindo a ordem do bit menos significativo ao bit mais significativo. Cada bit permanece um determinado tempo na linha de transmissão, e então, depois do último bit transmitido, o valor lógico 1 é novamente enviado afim de confirmar a transmissão de um dado. O protocolo descrito acima é designado na literatura como Não Retorno a Zero (NRZ) (Matic e Andric, 2000; Penido 2013). A Figura 7 representa o esquemático de uma unidade serie.

Figura 7 - Bloco de comunicação serial



Fonte: microcontroladores pic (2000)

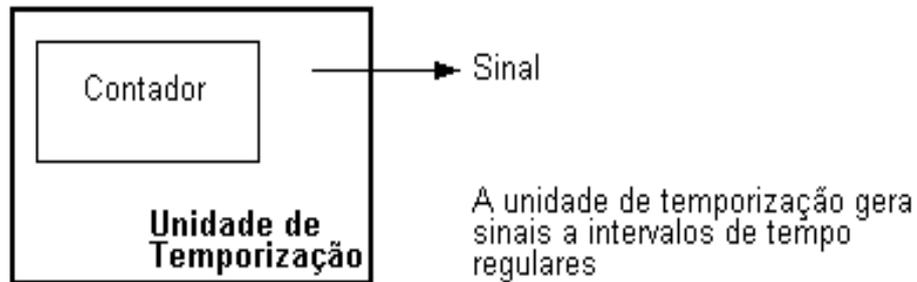
As linhas de trocas de dados trabalham de forma separada, sendo uma linha de envio e uma de recebimento, dessa forma sendo possível a troca simultânea de dados. O elemento responsável do microcontrolador que viabiliza essa comunicação é denominado como bloco de comunicação em série, que ao contrário a transmissão em paralelo, transmite o dado bit após bit sequencialmente. Quando um dado é recebido, esse é lido e armazenado na memória, já no caso da transmissão de um dado, o processo ocorre inversamente, onde determinado dado é carregado da memória para o local de transmissão através do barramento, e então enviado para uma unidade receptora, respeitando o protocolo escolhido (Matic e Andric, 2000; Penido 2013).

### 2.1.8 MICROCONTROLADORES: TIMER

O temporizador (timer) de um microcontrolador trabalha como um contador (Figura 8) criando um registro que aumenta em uma unidade por vez em um dado período de tempo fixo. Sabendo-se o valor total de unidades do contador dentro de um intervalo de tempo  $T_1$  e  $T_2$  e obtendo sua diferença, temos o tempo decorrido em determinado processo. O elemento temporizador (ou bloco de temporização) é

bastante relevante, pois através dele podemos coletar informações como hora, duração, protocolo, etc. (Matic e Andric, 2000; Penido 2013).

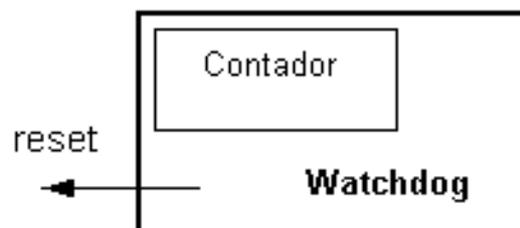
Figura 8 - Contador



Fonte: microcontroladores pic (2000)

A fluência pela qual o programa é executado dentro do microcontrolador também é determinada por um contador, denominado *Watchdog* (Figura 9). Esse contador trabalha de forma a resetar a programação no microcontrolador, quando esta trava por algum mal funcionamento ou por interferência externa. A cada ciclo correto da programação no microcontrolador, o *Watchdog* emite um sinal de valor lógico 0, afirmando a boa funcionalidade, em caso contrário, emite um sinal de valor lógico 1, assim resetando a programação (Matic e Andric, 2000; Penido 2013).

Figura 9 - Watchdog



Fonte:microcontroladores pic (2000)

### 2.1.9 MICROCONTROLADORES: CONVERSOR ANALÓGICO-DIGITAL

Ao trabalhar-se com periféricos é preciso utilizar um bloco novo, denominado conversor analógico-digital (A/D) (Figura 10), pois tais periféricos trabalham com sinais diferentes ao do microcontrolador. O conversor A/D é responsável pela conversão de um valor analógico em um valor binário e pelo caminho do mesmo dentro da CPU, afim que o microcontrolador possa entender e processar tal valor (Matic e Andric, 2000; Penido 2013). A figura a seguir mostra um conversor analógico-digital.

Figura 10 - Conversor analógico digital



Fonte: microcontroladores pic (2000)

### 2.1.10 MICROCONTROLADORES: PROGRAMA

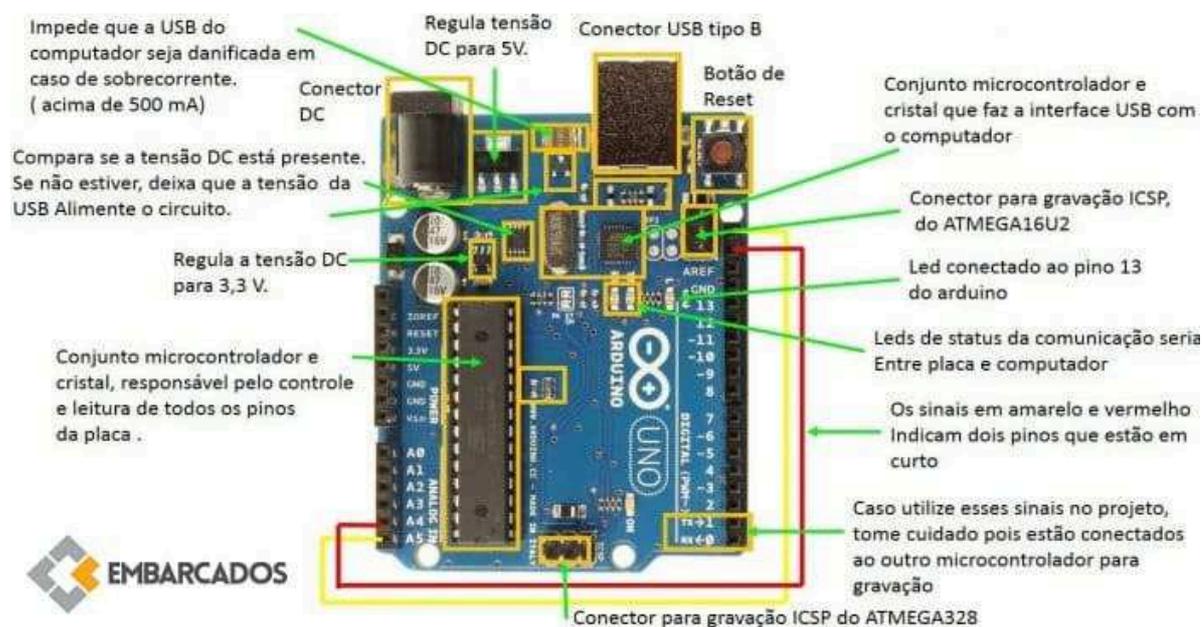
O programa tem como função orientar a funcionalidade do microcontrolador, dando-lhe as diretrizes a serem operadas. Um microcontrolador é construído afim de armazenar, transmitir, receber e operar um dado, mas como isso será feito, em que ordem e quais operações serão atribuídas a determinado dado, e determinado pela

programação. Existem vários tipos de linguagens de programação, contudo as mais utilizadas na programação de microcontroladores são *Assembly*, *C* e *Basic*. O que determina qual linguagem será usada, não depende somente de sua facilidade de programar, deve-se levar em conta também a velocidade de execução do programa, a quantidade de memória disponível e a complexidade do argumento ou dado a ser operado. A linguagem *Assembly* é a que detém maior velocidade de execução do programa, ocupando um espaço menor em relação as linguagens *C* e *Basic*, que por sua vez são mais fáceis de se programar (Matic e Andric, 2000; Penido 2013).

### **2.1.11 MICROCONTROLADORES: ARDUINO**

O arduino (Figura 11) é uma placa microcontroladora de baixo custo que viabiliza inúmeros projetos de controle e automação. Teve seu início na *Interaction Design Institute* na cidade de Ivrea, em 2005. Foi originalmente idealizada pelo professor Massimo Banzi, que buscava um meio de baixo custo que tornasse mais fácil para seus alunos de design trabalhar com tecnologia. Ao discutir seu problema com David Cuartielles, pesquisador da Universidade de Malmö, na Suécia, que buscava soluções semelhantes à de Banzi, e devido ao alto custo dos produtos oferecidos em mercado, Banzi e Cuartielles desenvolveram um microcontrolador que essencialmente tinha como exigências ser barato e que fosse de fácil manuseio, afim de que qualquer pessoa pudesse usá-lo, visando que seus alunos de arte e design utilizassem em seus projetos. Cuartielles desenhou a placa enquanto um aluno de Banzi, David Mellis, programou o software de execução da placa. Banzi contratou um engenheiro, Gianluca Martino, para que este confeccionasse uma tiragem de 200 placas. As placas eram vendidas em forma de kits para os estudantes e levavam o nome de Arduino em referência a um bar local frequentado por alunos e docentes. Sendo um sistema de fácil e utilização e baixo custo logo alcançou notada popularidade, sendo procurado por artista e designers de outros locais, assim sua versão original fora revisada, surgindo novas versões da placa, ampliando assim suas utilizações, sendo hoje um dos microcontroladores mais utilizados tanto para fins didáticos quanto para fins profissionais (McRoberts, 2011).

Figura 11 - Resumo placa Arduino



Fonte: embarcados (2011)

## 2.2 COMANDO NUMÉRICO COMPUTADORIZADO

### 2.2.1 COMANDO NUMÉRICO COMPUTADORIZADO: HISTÓRIA

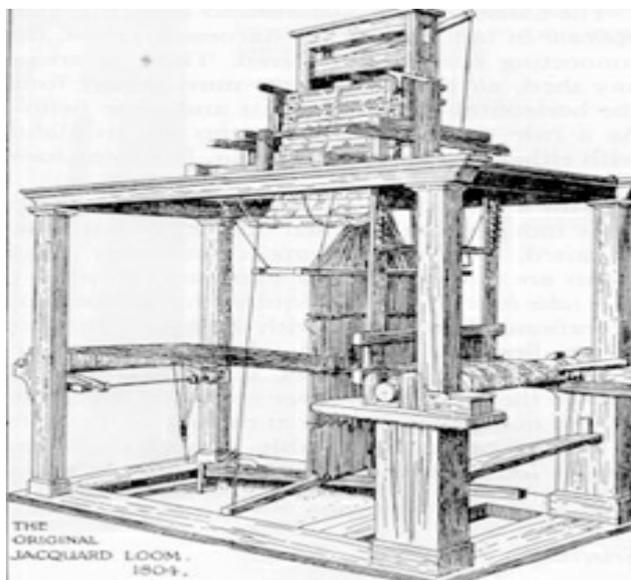
James Watt, inventor da máquina a vapor, iniciou a revolução com sua criação, com isso ele criou, também, a necessidade das ferramentas utilizadas pelas indústrias. Sem um equipamento que pudesse usinar um cilindro com uma tolerância mínima razoável, ficaria impossível a construção de motores. Para a construção de tal motor a vapor, foi necessária a invenção do ferreiro John Wilkinson, uma máquina que fazia cilindros fixando a parte exterior da peça em uma barra que era estendida longitudinalmente e um cabeçote de furação que se move devagar pela barra, fazendo assim a remoção do material (Silveira, 2007).

### 2.2.1.1 HISTÓRIA: DESENVOLVIMENTO DOS PRIMEIROS CONTROLES

Outra invenção necessária foi à criação de máquinas que possuem números em sua programação, em 1700, M. Falcon criou uma máquina de tricotar que utilizava-se de um cartão perfurado para movimentar as agulhas, assim fazendo a máquina tricotar automaticamente. Os cartões eram furados na sincronia do movimento, cada furo atuava como um acoplador, impulsionando assim uma agulha associada a uma coluna, esse mecanismo era necessário para puxar a linha colorida fazendo assim o tricô. Tal invenção foi limitada à baixa produção (Silveira, 2007).

Em 1807, o mecânico Joseph-Marie Jacquard melhorou o projeto de M.Falcon associando ao equipamento alta produtividade e baixo custo agregado ao produto final, esse método ainda é utilizado atualmente (Silveira, 2007). A Figura 12 ilustra a máquina de Jacquard.

Figura 12 - A máquina de tecer de Jacquard



Fonte: mundocnc (2008)

### **2.2.1.2 HISTÓRIA: DESENVOLVIMENTO DOS COMPUTADORES**

O primeiro projeto de computador digital foi concebido por Charles Babage, em 1800, desenvolvido, porém nunca concluído. Em 1945 o exercito dos Estados Unidos criou o ENIAC, uma máquina lenta e que não possuía nenhum programa em sua memória, e as sequências de informações deveriam ser inseridas externamente.

No período da segunda guerra mundial, o matemático John Von Neumann criou o computador digital com programas em memória, tornando-se assim um equipamento de extrema importância para o comercio e indústria complexa. (Silveira, 2007).

### **2.2.1.3 HISTÓRIA: LINGUAGEM G**

A linguagem G foi criada em 1958 no MIT e foi adotada como padrão pelos fabricantes de comandos, nessa padronização rígida para comandos básicos e universais, dando liberdade para as funções de recurso próprio do fabricante. Com a possibilidade de funções diferentes, dentro de uma mesma linguagem, a criatividade entre fabricantes se tornou um diferencial no projeto de maquinas de comando.

O programa CN (Comando Numérico) foi uma maneira criada para o homem se comunicar com a máquina através de códigos, transformando um desenho ou uma peça em números e letras, transformando-a em pura matemática. A seqüência de caracteres é memorizada pelo comando sendo executadas, uma a uma (Silva, 2002).

### **2.2.2 COMANDO NUMÉRICO COMPUTADORIZADO: VANTAGENS**

O objetivo de um CNC é o ganho de tempo e produtividade em uma linha de produção, é possível ter o mesmo resultado utilizando máquinas convencionais, porém, é um processo mais trabalhoso e a repetição das peças produzidas faz com que, em alguns casos, se torne inviável tal produção. A automação é de longe a principal vantagem do CNC, com a precisão do equipamento, os erros humanos são reduzidos ou totalmente descartados. Há alguns ciclos de usinagem que não é necessário à presença de um operador, tendo o trabalhador mais tempo para realizar outras funções. Entre os benefícios que o CNC proporciona ao usuário, podemos citar menor fadiga, eliminação do erro humano e a previsão de término do ciclo de usinagem (Neves, 2005; Paredes, 2013; Sanches 2009).

Outro benefício que se destaca, é uma produção de peças com uma precisão muito alta, mesmo em uma produção repetitiva, a precisão é mantida para um número elevado de peças produzidas (Neves, 2005; Paredes, 2013; Sanches 2009).

Por último, a flexibilidade, onde é possível facilmente trocar o programa para alterar o corte a ser realizado na peça, com a mudança no programa se altera toda a usinagem que deveria ser realizada, nos dias de hoje é de grande importância tal benefício, pois o mercado exige agilidade nos processos de fabricação (Neves, 2005; Paredes, 2013; Sanches 2009).

### **2.2.3 COMANDO NUMÉRICO COMPUTADORIZADO: FUNDAMENTOS DO CONTROLE NÚMERO**

O controle numérico é a forma como a máquina CNC enxerga as dimensões e toda geometria da peça a ser trabalhada, conseguimos essas informações a partir de cotas em desenhos técnicos. Com o CNC computando essas informações é possível realizar a automação da usinagem.

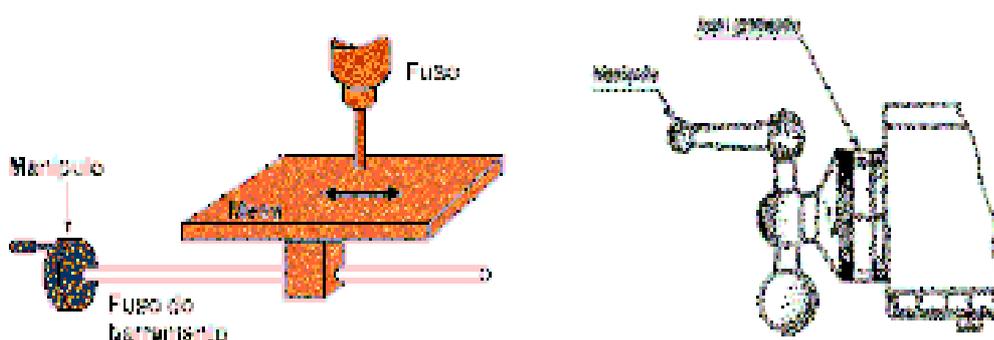
O Comando Numérico Computadorizado (CNC) é um equipamento que recebe a informação de como a máquina deveria operar e através da sua

programação essa informação é processada e devolvida no formato de impulsos elétricos para os atuadores, os atuadores (motores) fazem toda a seqüência de movimentos necessários para que a usinagem seja realizada (Neves, 2005; Paredes, 2013; Sanches 2009).

### 2.2.3.1 FUNDAMENTOS DO CONTROLE NUMÉRICO: CONTROLE DOS MOVIMENTOS

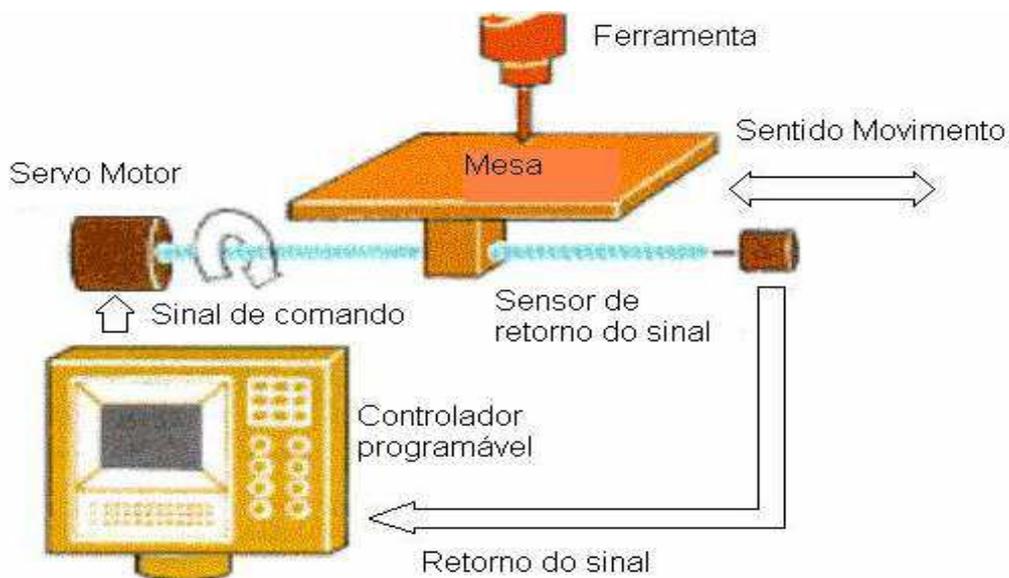
O movimento dos motores é realizado em eixos, todos os CNC's possuem de duas ou mais direções de movimento, ou seja, dois ou mais eixos. Os eixos devem ser posicionados para que haja os movimentos de translação. Os movimentos realizados são rápidos, lineares e circulares e esses tipos de movimento são programáveis junto com a quantidade de movimento a ser realizado e a taxa de avanço, também conhecido como "*feed rate*". Nas máquinas convencionais (Figura 13) os movimentos eram feitos manualmente por meio de manivelas, no CNC os movimentos são realizados através de servos motores ou motores de passo (Figura 14) (Neves, 2005; Paredes, 2013; Sanches 2009).

Figura 13 - O movimento da mesa de uma máquina convencional



Fonte: mundocnc (2008)

Figura 14 - Sistema de acionamento de uma máquina CNC



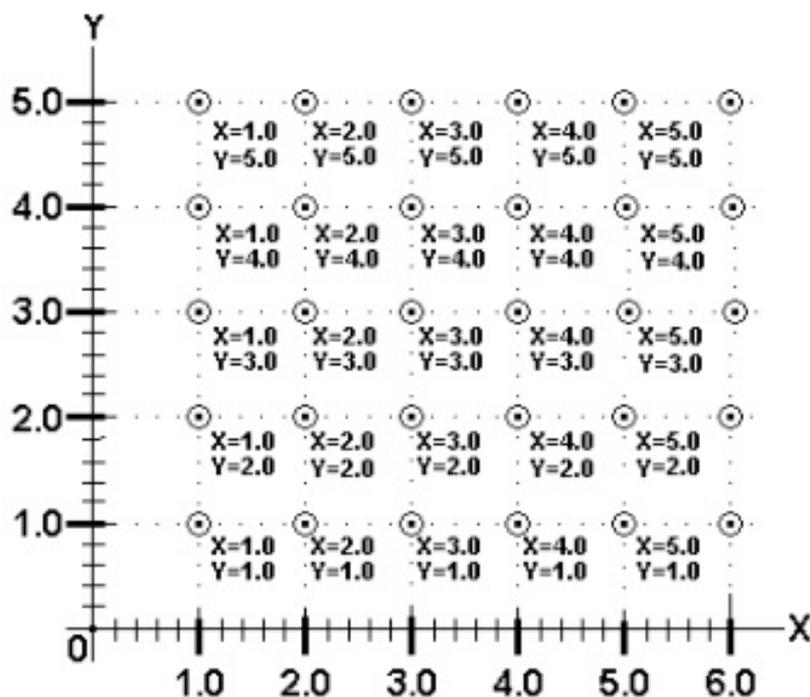
Fonte: mundocnc (2008)

### 2.2.3.2 FUNDAMENTOS DO CONTROLE NUMÉRICO: SISTEMAS DE COORDENADAS

A forma que o CNC comanda os movimentos a serem realizados é baseada em coordenadas, as coordenadas mais utilizadas pelo controle do CNC são as coordenadas cartesianas, podem ser chamadas também de coordenadas retangulares (Figura 15), ou podem ser utilizadas as coordenadas polares (Sanches 2009).

O ponto de origem do movimento é chamado, normalmente, de ponto zero, os demais pontos são programados pelo operador, a fim de obter o desenho final, cada eixo do sistema pode ser comparado a uma linha simples de um gráfico, e os eixos de movimento são divididos por incrementos de medida, no padrão métrico é utilizado o milímetro com divisões de milésimos de milímetros, no padrão inglês, as subdivisões do eixo são dadas em 0.0001 de polegadas, para eixos rotativos, tal incremento de medida chega a 0.001 graus (Neves, 2005;).

Figura 15 - Sistema de coordenadas retangular para o plano XY

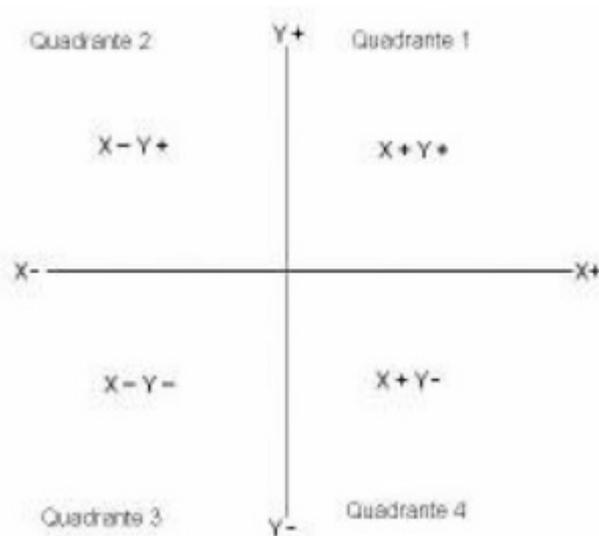


Fonte: mundocnc (2008)

No caso da Figura 15 temos os eixos X e Y embora para cada CNC pode haver variação do nome dos eixos de acordo com o tipo da maquina, os eixos podem ser chamados de Z, A, B, C, U, V e W. O operador determina o zero do eixo, geralmente o ponto zero é escolhido onde se iniciam as graduações de cada eixo (Neves, 2005; Paredes, 2013; Sanches 2009).

Algumas CNC's trabalham também em quatro quadrantes (Figura 16), nesse caso são utilizadas coordenadas negativas.

Figura 16 - Quadrantes



Fonte: mundocnc (2008)

### 3 METODOLOGIA

Ao idealizar o projeto de uma CNC de baixo custo, é ideal também, que seja de fácil montagem. Os materiais utilizados são encontrados em comércio local e os elementos de união seguem o mesmo padrão baixo custo e facilidade de operação

#### 3.1 MATERIAIS

- Tubos de alumínio 25cm X 1 cm
- 2 Duratex 40 x 40 cm
- 3 Duratex 6 x 10 cm
- 1 Duratex 15 x 12 cm
- 4 Sarrafos 4 x 2 x 20 cm
- 2 Motores de Passo modelo 28BYJ - 48 5v

- 2 Drivers para motor de passo modelo ULN2003
- 1 Servo Motor Towerpro Sg90
- 1 Arduíno R3 UNO
- Protoboard
- Cabos para Jumper
- Linha de Nylon para pesca
- Cola tipo Adesivo Instantâneo
- Retalhos de madeira
- Suportes plásticos para prateleira
- Arame
- Seringa de insulina
- Percloroeto de Ferro

### **3.2 MONTAGEM**

A montagem do projeto é de simples execução, também realizado com materiais de fácil acesso.

Figura 17 - Materiais para estrutura



Fonte: Elaborado pelo Autor (2017)

Primeiro é fixado os tubos de alumínio nos sarrafos com três furos, serão montados dois tubos para cada dois sarrafos (Figura 17), esses serão os guias para a base da CNC e para a base da caneta, a fixação é realizada com cola do tipo adesivo instantâneo. Para as guias inferiores o espaçamento é maior, pois apoiara a base para o processo de fresamento. As guias superiores movimentarão a caneta sobre a placa de cobre, portanto devera ter uma distancia menor entre eixos. Os sarrafos com os tubos são fixado nas duas placas Duratex de 40 x 40, todas as fixações são feitas com a cola adesiva.

Figura 18 - Estrutura Montada

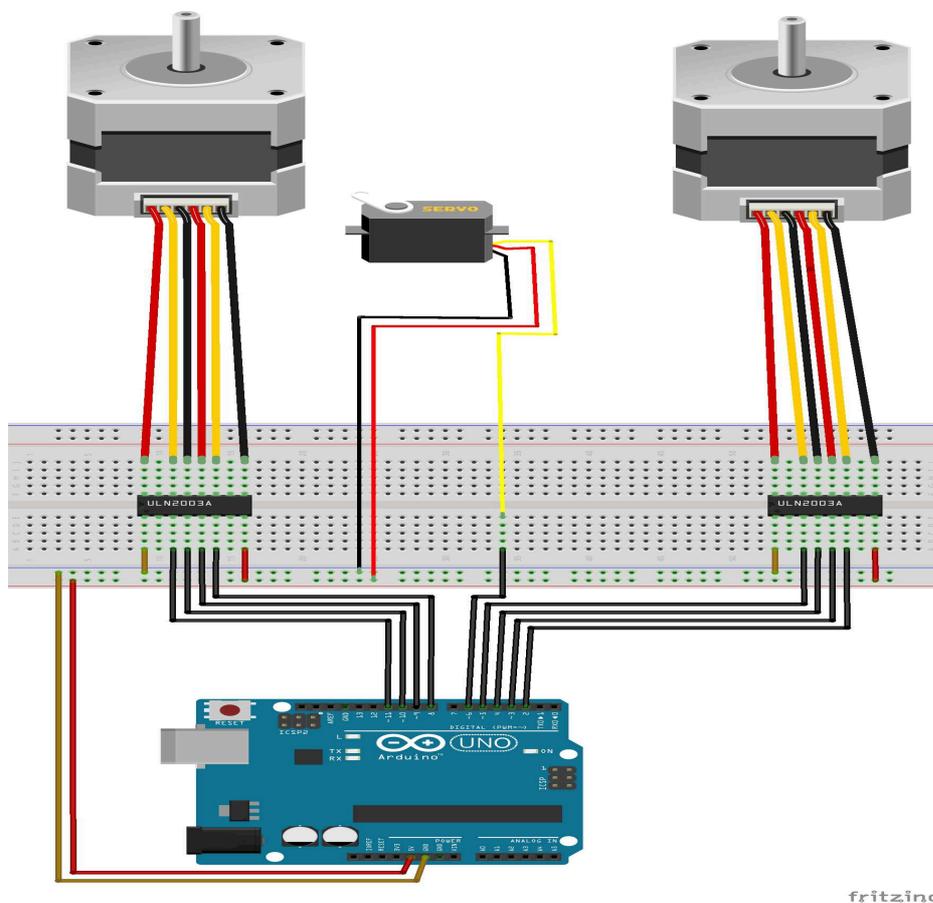


Fonte: Elaborado pelo Autor (2017)

Próximo passo é a fixação dos elementos eletromecânicos, compostos dos motores de passo e os driver's dos motores juntamente com a protoboard na base da estrutura

As ligações elétrica entre motores e drivers são realizadas utilizando os cabos jumpers, para a ligação dos dois drivers dos motores e o microcontrolador, é utilizado a protoboard, pois as entradas disponíveis no mesmo são limitadas.

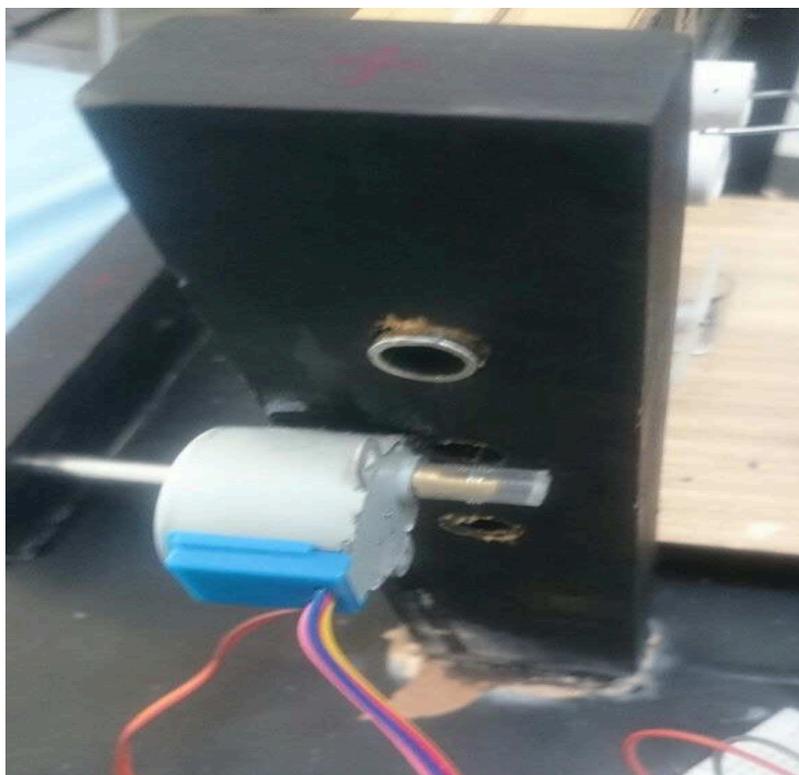
Figura 19 - Esquemático do Fritzing



Fonte: Elaborado pelo Autor (2017)

As linhas de pescas são utilizadas nos dois motores, para realizar a movimentação da base e caneta, no motor inferior elas realizam a movimentação da base do CNC, no motor superior elas movimentam a caneta, podemos utilizar do plano cartesiano para o melhor entendimento dos movimentos realizados, o motor inferior (base) descreve seu movimento no eixo X e o movimento da caneta descreve seu movimento no eixo Y. A união da linha com a base inferior e com o servo motor é feita utilizando de papel e cola adesiva, nos eixos dos motores (Figura 20) de passo é colocado um segmento do tudo da seringa para deixar a rotação da linha mais equilibrada, pois o eixo dos motores não é cilíndrica, os dois chanfros no eixo prejudica a precisão da impressão

Figura 20 - Eixo do motor



Fonte: Elaborado pelo autor (2017)

A base de impressão é fixada na linha de pesca utilizando cola e papel, na parte inferior da base é utilizado pequenos retalhos retangulares de madeira e os suporte plástico para guiar o movimento da base nos tubos de alumínio, assim limitando o movimento da base apenas no eixo, tirando assim a possibilidade de movimento indesejado que poderia comprometer a impressão. A base menor vertical foi furada para que o servo motor seja fixado, a caneta foi fixada por meio de plásticos de embalagem, arame e os suportes de plástico, que foram utilizados para guiar o movimento da caneta e também para transferir o movimento do servo motor para a caneta. O servo levanta a caneta de impressão, quando inicia a impressão o servo libera a caneta e que por gravidade entra em contato com a placa de cobre.

Figura 21 - Base de fixação do servo motor e caneta



Fonte: Elaborado pelo autor (2017)

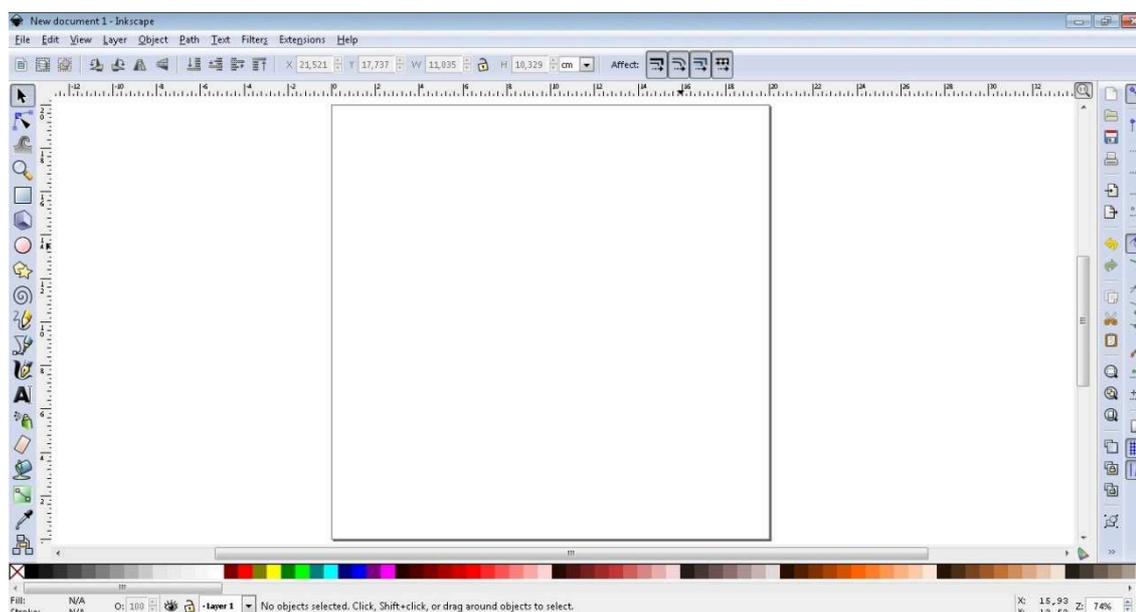
### 3.3 GERAÇÃO DO G-CODE (VETORIZAÇÃO DA IMAGEM)

Para obtenção das imagens em forma vetorial (G-Code) é utilizado o programa de processamento de imagens Inkscape e uma extensão desenvolvida para CNC's caseiras denominado MakerbotUnicorn G-Code, sendo esta compatível com o Inkscape versão 0.48.5. As imagens a seguir demonstram a obtenção do G-Code através dos softwares citados.

### 3.3.1 INKSCAPE 0.48.5

Software de edição de imagens e desenho gratuito em sua versão compatível a extensão geradora de C-Code.

Figura 22 - Software de edição de imagem e desenho Inkscape

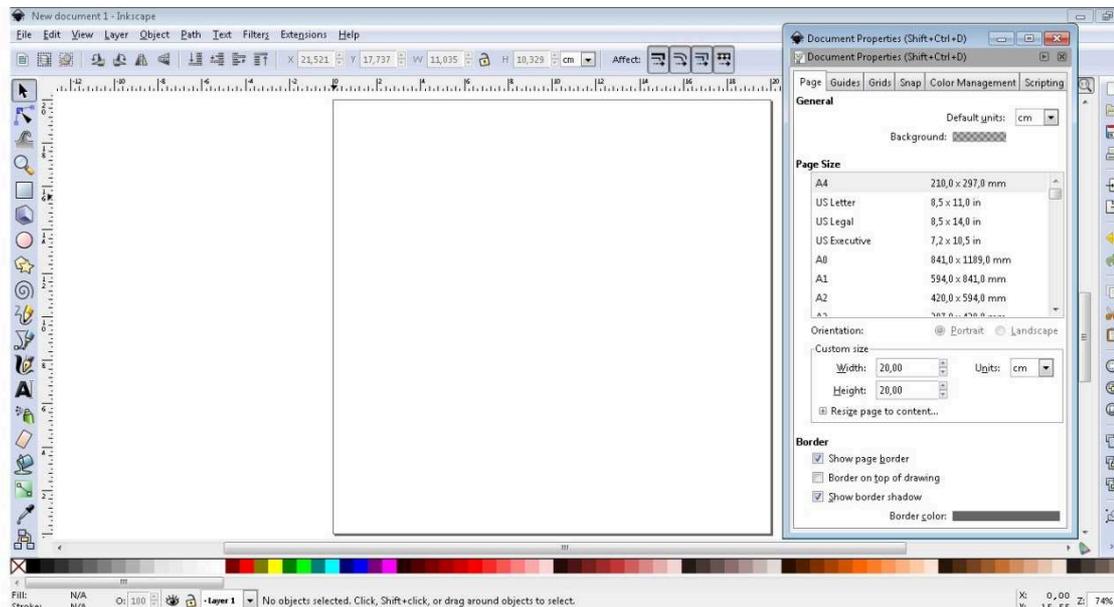


Fonte: Printscreen do aplicativo Inkscape (2017)

#### 3.3.1.2 CONFIGURAÇÃO DA ÁREA DE IMPRESSÃO

Para se configurar a área de impressão (Figura 23), deve-se ir em File>DocumentProperties ou CTRL+Shift+D. Em **General** deve-se mudar DefaultUnits para centímetros e em **PaperSize** deve-se mudar para centímetros e configurar a página para 20x20 cm e onde a área útil de desenho será entre 10 a 20 cm, pois o programa entende que, essa área de 20x20 cm, vai de -10 a 10 cm.

Figura 23 - Configuração da Área de Impressão

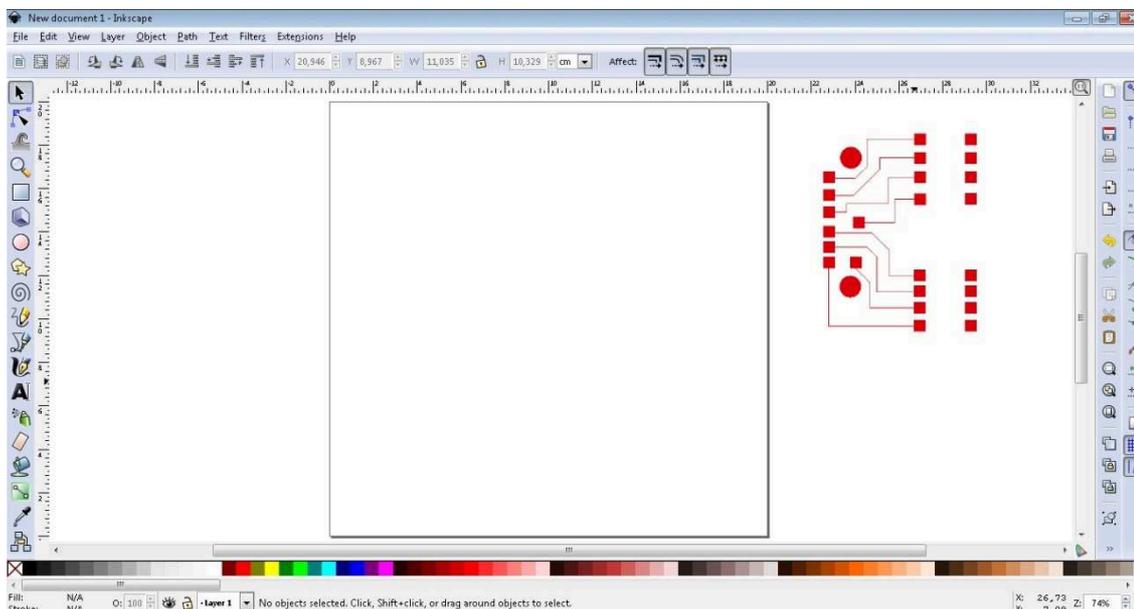


Fonte: Printscreen do aplicativo Inkscape (2017)

### 3.3.1.3 CARREGAMENTO DO DESENHO

Para se carregar o desenho ou projeto que se deseja imprimir deve-se ir File>Open, CTRL+O ou somente selecionar e carregar o desenho sobre o ícone do Inkscape (Figura 24).

Figura 24 - Carregamento do Desenho

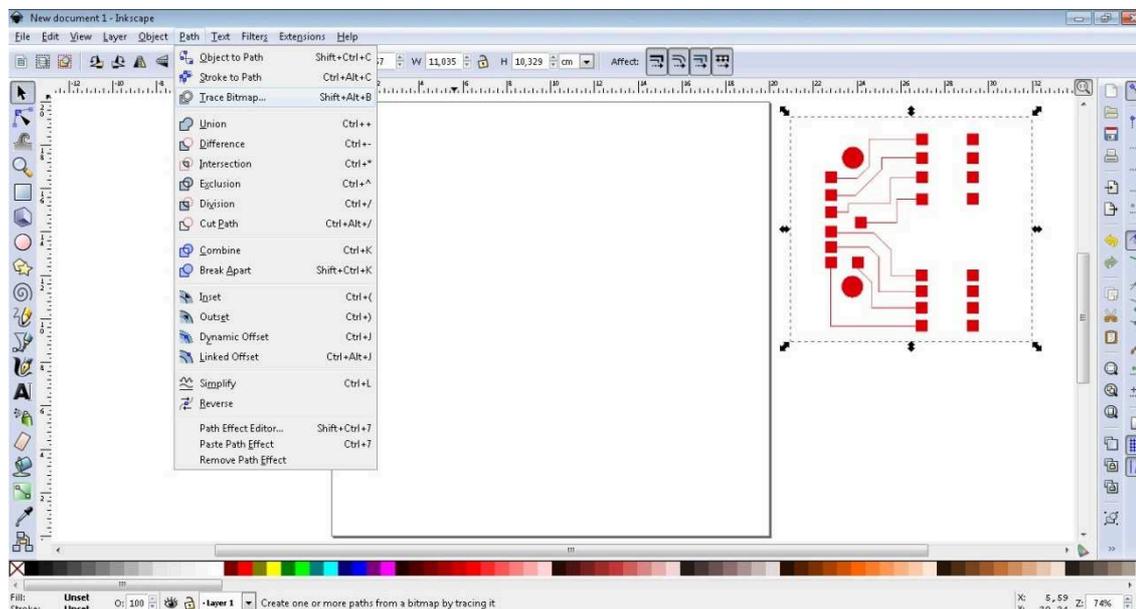


Fonte: Printscreen do aplicativo Inkscape (2017)

### 3.3.1.4 EXTRAINDO CONTORNO OU BORDAS PARA IMPRESSÃO

Para obtenção das bordas ou contornos para impressão (Figura 25) deve-se ir em Path>Tracebitmap.

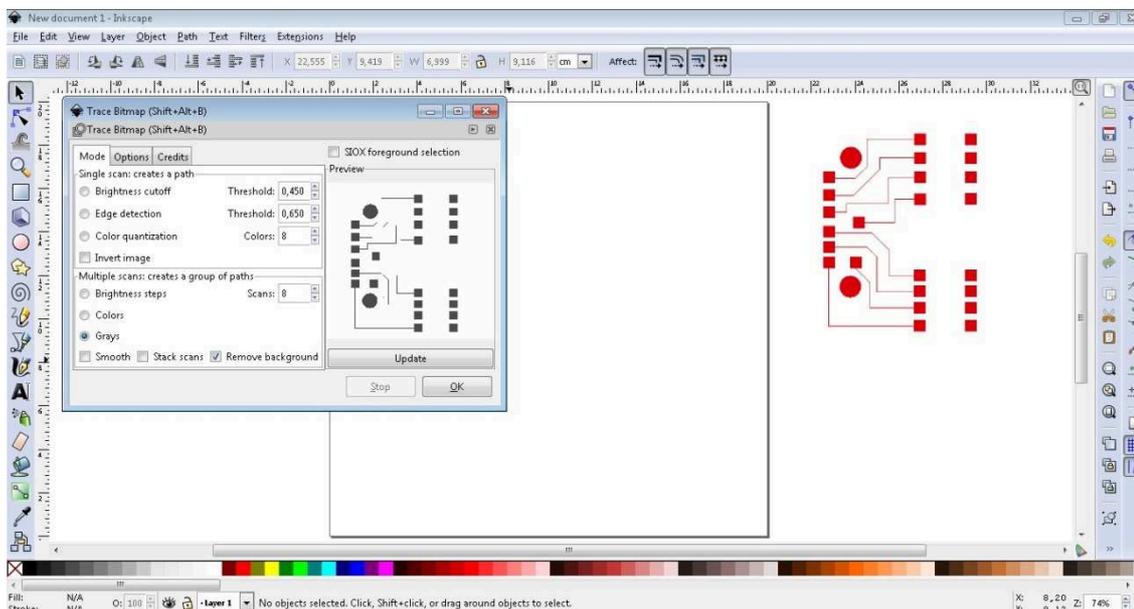
Figura 25 - Extração Contorno ou Bordas



Fonte: Printscreen do aplicativo Inkscape (2017)

Em Tracebitmap deve deixar ticado apenas a opção de Remove Background e Grays (Figura 26), a fim de se gerar uma segunda imagem sem o fundo branco.

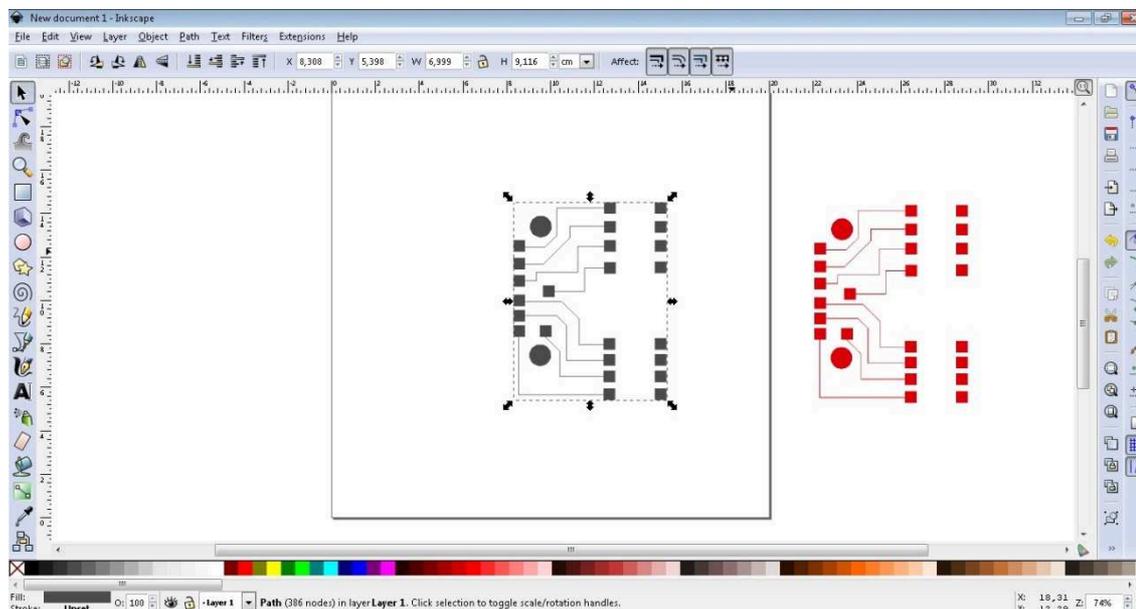
Figura 26 - Remoção do Background



Fonte: Printscreen do aplicativo Inkscape (2017)

A imagem com fundo branco (vermelha) é compreendida pelo programa como imagem sólida, não conseguindo distinguir os contornos que devem ser impressos para gerar os nós de coordenadas. A imagem sem fundo (cinza) é compreendida apenas pelos contornos desejados (Figura 27), sendo esses pontos espaciais localizáveis pelo programa, podendo assim gerar os nós de coordenadas.

Figura 27 - Imagem Cinza sem fundo e Imagem Vermelha com fundo

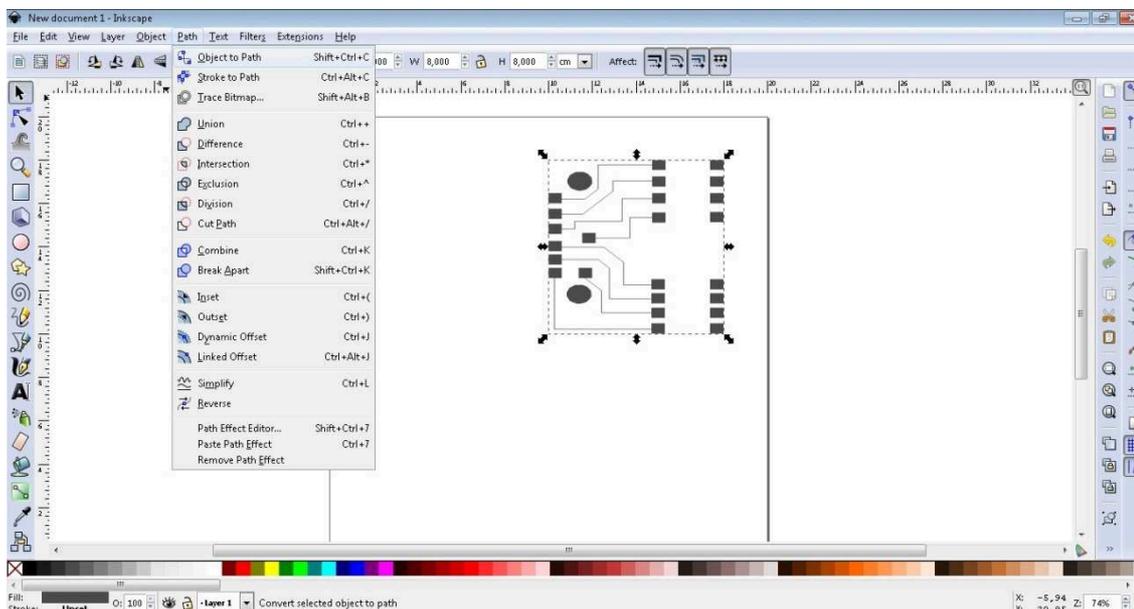


Fonte: Printscreen do aplicativo Inkscape (2017)

### 3.3.1.5 GERANDO OS NÓS DE COORDENADAS

Para obtenção dos nós que possibilitam a vetorização da imagem (Figura 28), deve-se ir em Path>Object Path ou CTRL+Shift+C e clicar com o botão esquerdo no mouse.

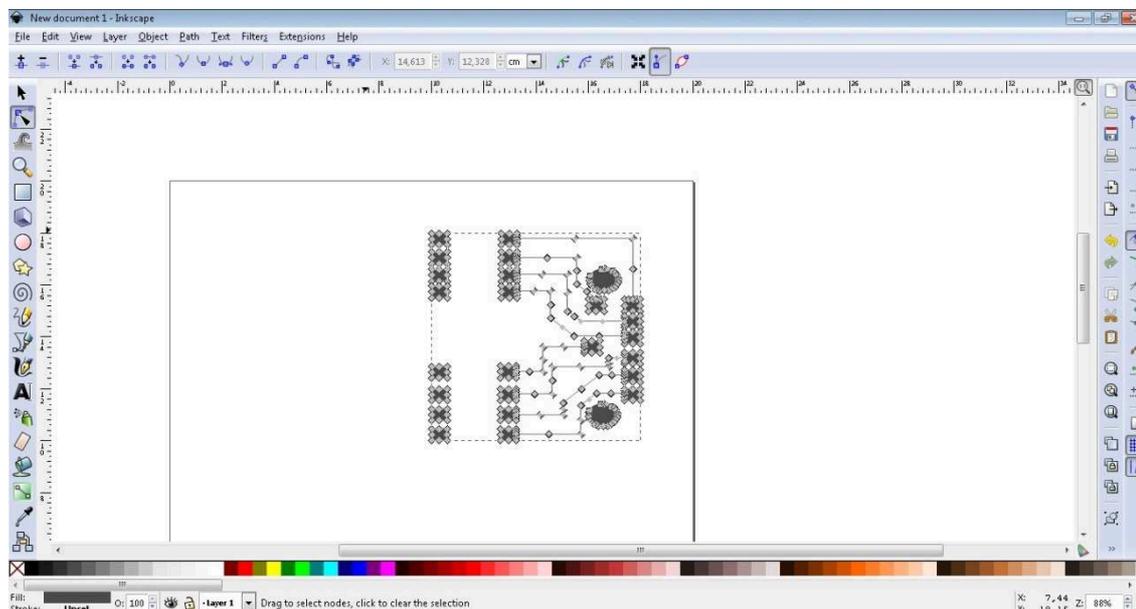
Figura 28 - Vetorização da Imagem



Fonte: Printscreen do aplicativo Inkscape (2017)

Ao clicar na imagem, os nós serão gerados. Esses nós representam posições espaciais orientadas em coordenadas X e Y (Figura 29). O desenho deve estar posicionado dentro da área entre 10 a 20 cm, assim gerando coordenadas dentro do quadrante positivo da área de desenho.

Figura 29 - Coordenadas Geradas em X e Y

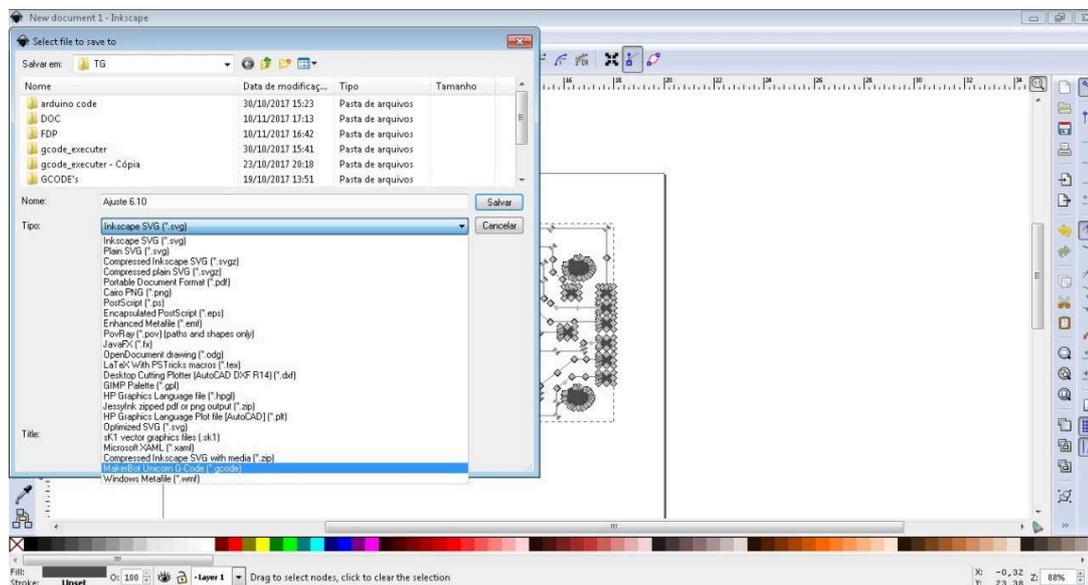


Fonte: Printscreen do aplicativo Inkscape (2017)

### 3.3.1.6 GERANDO O G-CODE

Após a geração dos nós, para se obter o G-Code (Figura 30) deve-se ir em File>Save As e salvar como extensão MakerbotUnicorn G-Code.

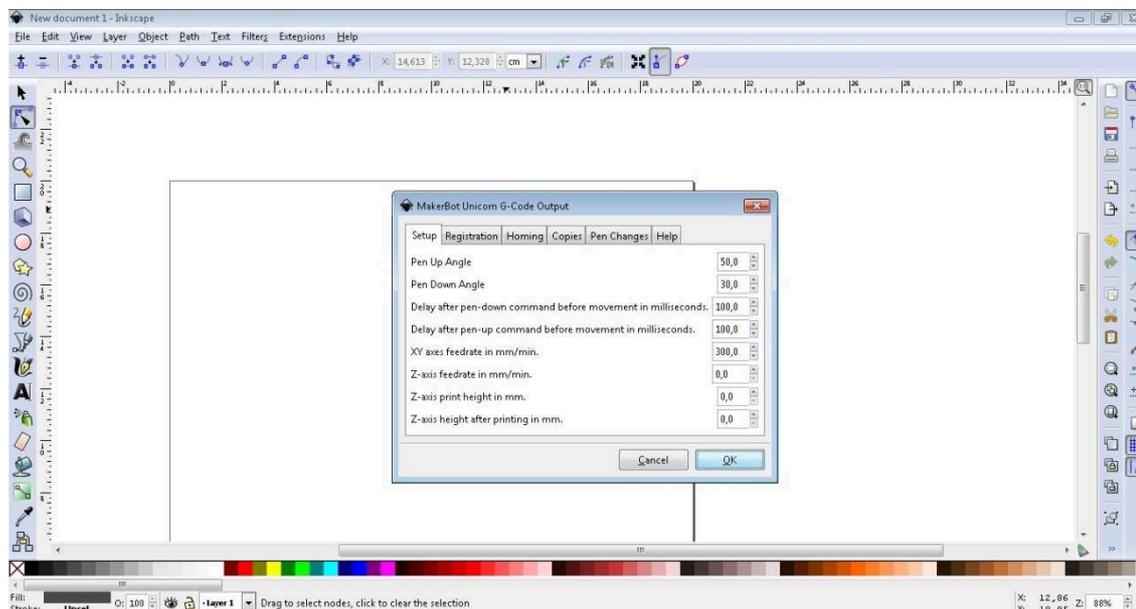
Figura 30 - Gerando Arquivo de G-Code



Fonte: Printscreen do aplicativo Inkscape (2017)

Após salvar na extensão.gcode o menu da figura abaixo é ativado, onde é possível se configurar a angulação do servo que controla a caneta, o tempo de espera do servo antes de implementar o comando de subida ou descida do servo e a velocidade de impressão.

Figura 31 - Configuração de Impressão

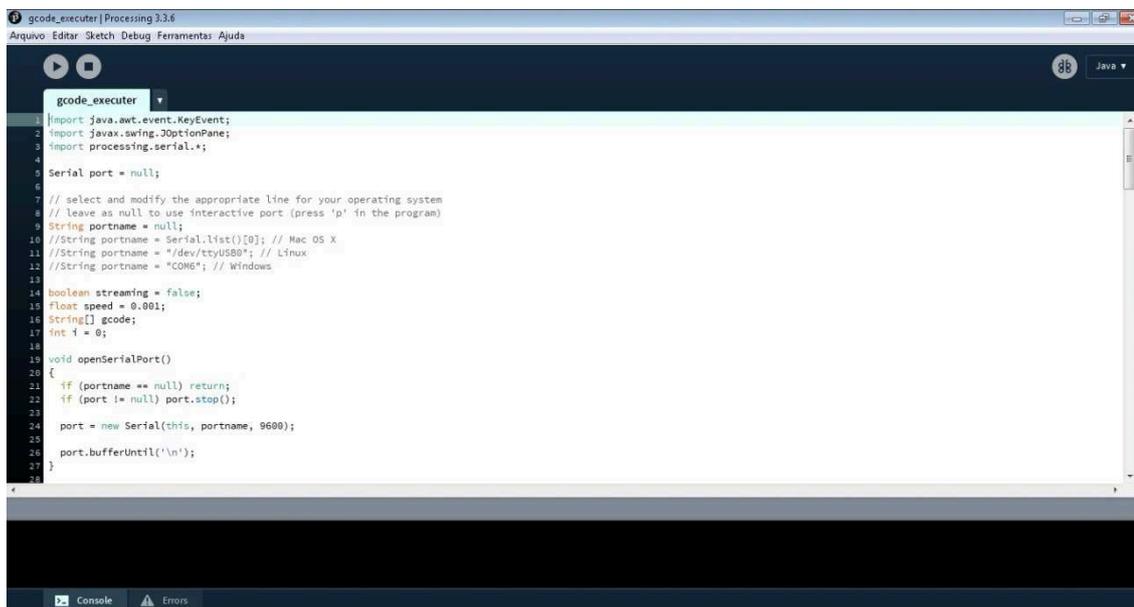


Fonte: Printscreen do aplicativo Inkscape (2017)

### 3.3.2 - EXECUTANDO O G-CODE

Para a execução dos G-Codes gerados para impressão, foi utilizado uma linha de programação pronta que é executada em Java através da IDE (Figura 32) (IntegratedDevelopmentEnvironment ou Ambiente de Desenvolvimento Integrado) Processing.

Figura 32 - IDE Processing

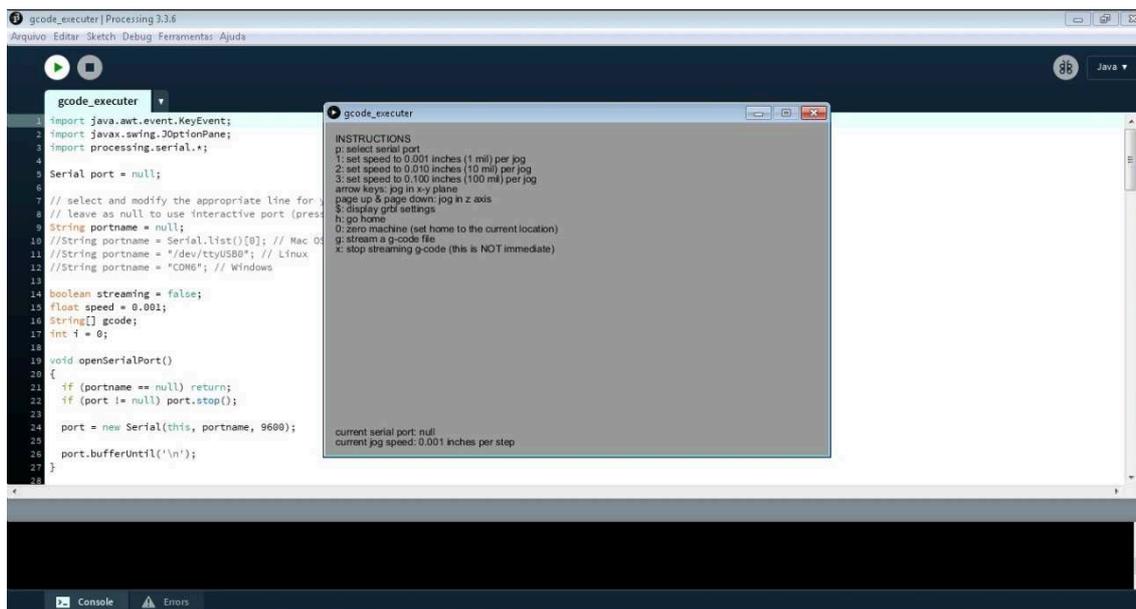


```
gcode_executor
1 import java.awt.event.KeyEvent;
2 import javax.swing.JOptionPane;
3 import processing.serial.*;
4
5 Serial port = null;
6
7 // select and modify the appropriate line for your operating system
8 // leave as null to use interactive port (press 'p' in the program)
9 String portname = null;
10 //String portname = Serial.list()[0]; // Mac OS X
11 //String portname = "/dev/ttyUSB0"; // Linux
12 //String portname = "COM6"; // Windows
13
14 boolean streaming = false;
15 float speed = 0.001;
16 String[] gcode;
17 int i = 0;
18
19 void openSerialPort()
20 {
21   if (portname == null) return;
22   if (port != null) port.stop();
23
24   port = new Serial(this, portname, 9600);
25   port.bufferUntil('\n');
26 }
27
28
```

Fonte: Printscreen do aplicativo Inkscape (2017)

Executando a programação no Processing, uma janela é iniciada e essa representa a interface de controle da CNC (, onde é possível carregar o G-Code anteriormente obtido. Para carregar o G-Code, é preciso primeiramente alinhar essa programação com a mesma porta USB de execução do Arduino (no nosso caso COM 3) e para isso deve se usar a tecla **P** e selecionando a porta em USB em uso. Após isso já se pode iniciar a impressão do G-Code utilizando a tecla **G**, abrindo uma janela onde é possível buscar o diretório onde esta salvo seu G-Code anteriormente feito. Contudo a interface em questão ainda proporciona algumas outras configurações e funcionalidades, como variação da velocidade de impressão, movimentação dos motores através do direcionais, botão para parar a impressão, etc.

Figura 33 - Interface de Controle da CNC



Fonte: Printscreen do aplicativo Inkscape (2017)

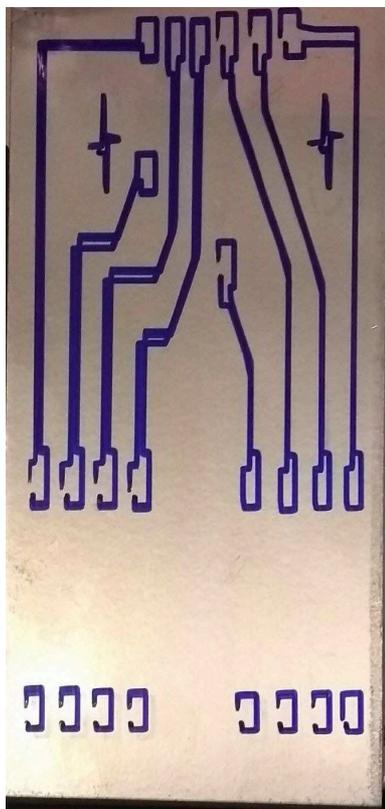
## 4 RESULTADOS

Com o projeto realizado observamos que é possível criar uma CNC, de fácil montagem utilizando materiais acessíveis para a criação de placas de circuito impresso. A CNC gera com precisão para os desenhos dos circuitos comparados a placas onde o circuito seria desenhado manualmente, evitando assim erro humano. Também é possível perceber que o projeto facilita a intercambialidade dos desenhos dos circuitos, devendo ser aplicado ao software para transformação em linguagem G.

Após a impressão do desenho do circuito na placa de cobre, a placa foi levada para o banho de percloroeto de ferro, onde todo o cobre que não possui a tinta foi corroído, mantendo assim somente as trilhas do circuito projetadas.

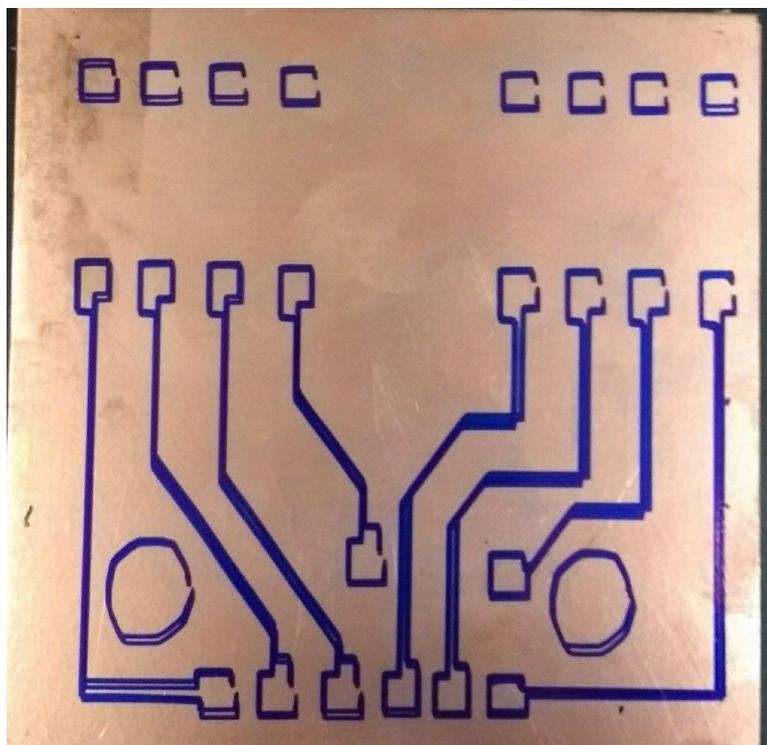
As dificuldades do projeto se apresentaram principalmente na fase de montagem da estrutura, muitas peças não tiveram uma perfeita união e nas peças moveis esse detalhe gerou movimentos indesejáveis o que atrapalhou a precisão do desenho, esse problema foi resolvido colocando elementos que limitassem os movimentos indesejados, foi preciso a montar e desmontar algumas partes varias vezes. Houve dificuldade também no uso do software para gerar o G-code, pois é preciso alinhar o desenho na impressão, o que foi solucionado com vários testes e tentativas. Ainda assim em alguns pontos é necessário um acabamento manual no desenho, a impressão tem algumas falhas e falta no preenchimento, esse problema é causado no processo de transformar o contorno do desenho em caminho, são falhas mínimas de fácil correção manual.

Figura 34 - Placa 5x10 Impressa pela CNC



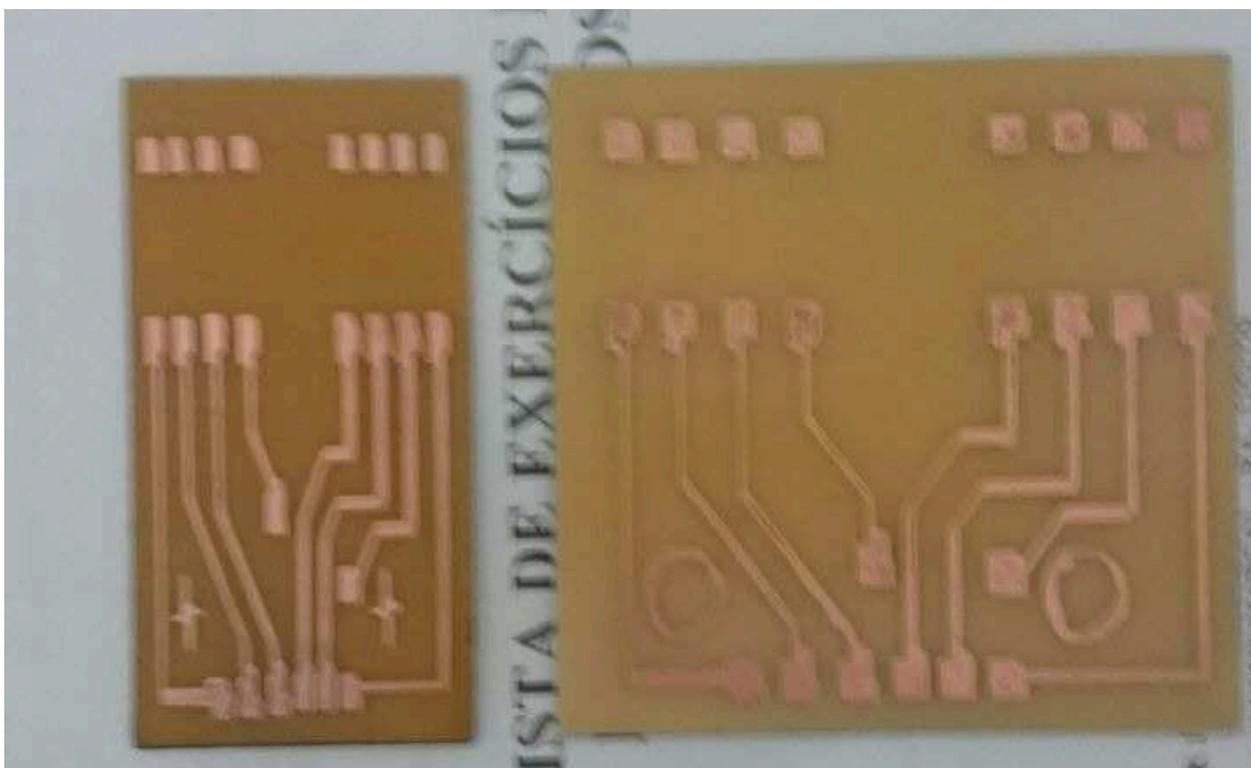
Fonte: Elaborado pelo Autor (2017)

Figura 35 - Placa 10x10 Impressa pela CNC



Fonte: Elaborado pelo Autor (2017)

Figura 36 - Placas 5x10 e 10x10 após tratamento químico com Percloroeto de Ferro



Fonte: Elaborado pelo Autor (2017)

## 5 CONCLUSÃO

Os projetos de circuitos impressos e suas complexidades necessitam da utilização de meios autônomos e precisos para sua implementação. Máquinas encontradas no mercado têm alto custo, tornando inviável a aquisição das mesmas para impressão de projetos pequenos e/ou caseiros. A fresa CNC desenvolvida no presente trabalho mostrou-se uma opção de baixo custo e eficiente para a obtenção de placas de circuito impresso, diminuindo o tempo de confecção e viabilizando a impressão de projetos complexos. Contudo, ainda existem possibilidades de melhora do projeto em questão, que podem ser implementado em projetos futuros, como a troca dos cabos de nylon por fusos, onde os passos do fuso garantirão maior precisão das impressões em um tempo menor. A adaptação do software de desenho ou a utilização de um novo software que garanta a conversão direto de um projeto *Gerber* para G-Code, assim garantindo a confiabilidade das dimensões do projeto de PCI (Placas de Circuito Impresso). A troca da caneta por uma pequena broca com o intuito de fazer *milling* (remoção direta do cobre da placa virgem durante a impressão), assim removendo a necessidade de ataque químico com percloro de ferro.

## 6 REFERÊNCIAS

ASSIS, Wânderson de Oliveira. **APLICAÇÕES DE MÁQUINAS-FERRAMENTA COM PROTOTIPAGEM RÁPIDA E “ENGENHARIA REVERSA”**. 2009. 3 f. Dissertação (Mestrado) - Curso de Automação e Controle, Centro Universitário do Instituto Mauá de Tecnologia, Centro Universitário do Instituto Mauá de Tecnologia, [s. L.], 2009. Disponível em: <<http://maua.br/files/artigos/artigo-maquinas-ferramenta-2.pdf>>. Acesso em: 19 05 2017.

**HOW TO MAKE AN EASY ARDUINO CNC PLOTTER part 1. [s. L.]: Easy Arduino Projects**, 2017. Son., color. Disponível em: <<https://www.youtube.com/watch?v=szXNpl4GydA>>. Acesso em: 19 de maio de 2017.

MATIC, Nebojsa; ANDRIC, Dragan. **Microcontroladores PIC**. São Paulo: Mansong, 2000. 252 p.

MCROBERTS, Michael. **Arduino Básico**. São Paulo: Novatec, 2011. 456 p.

**MUNDO CNC. Historico**. Disponível em: <<http://www.mundocnc.com.br/historico.php> > Acesso em: 25 de Maio de 2017

NEVES, José Antonio. **DESENVOLVIMENTO DE UM DISPOSITIVO ELETROMECAÂNICO PARA TROCA AUTOMÁTICA DE FERRAMENTAS EM CENTROS DE TORNEAMENTO CNC**. 2005. 115 f. Dissertação (Mestrado) - Curso de Engenharia Mecânica, Universidade de Taubaté, Taubaté, 2005.

OURO PRETO. Édilus de Carvalho Castro Penido. Professor. **Microcontroladores**. Ouro Petro: E-tec Brasil, 2013. 80 p.

PAREDES, Miguel Eduardo Gutierrez. **DESENVOLVIMENTO DE UMA FRESADORA CNC ADERENTE À NORMA STEP-NC BASEADO NO**

**CONTROLADOR DE MÁQUINA AVANÇADO (EMC2).** 2013. 227 f. Dissertação (Mestrado) - Curso de Engenharia Mecânica, Universidade de Brasília, Brasília, 2013.

SANCHES, José Matheus. **DESENVOLVIMENTO DE UMA FRESADORA CNC DE BAIXO CUSTO PARA FINS DIDÁTICOS.** 2009. 119 f. Dissertação (Mestrado) - Curso de Engenharia Mecânica, Universidade de São Paulo, São Carlos, 2009.

SILVA, Sidney Domingues da. Cnc - **Programação de Comandos Numéricos Computadorizados - Torneamento** - Série Formação Profissional. [s. L.]: Érica, 2002. 308 p.

SILVEIRA, Ricardo César Alves. **DESENVOLVIMENTO DE UM EQUIPAMENTO MECÂNICO COM CONTROLE NUMÉRICO COMPUTADORIZADO PARA PRODUÇÃO DE PROTÓTIPOS EM ESCALA.** 2007. 147 f. Dissertação (Mestrado) - Curso de Engenharia Mecânica, Universidade Federal de Minas Gerais, Belo Horizonte, 2007.