

Proposta de Integração Entre CAD e CNC para Máquina de Corte

Herysson Rodrigues Figueiredo¹, Alexandre de Oliveira Zamberlan¹, Reiner Frantesco Perozzo¹, Sylvio André Garcia Vieira¹

¹Sistemas de Informação – Centro Universitário Franciscano (UNIFRA)
97.010.032 – Santa Maria – RS – Brasil

heryssonf@gmail.com, {alexandre, reiner, sylvio}@unifra.edu.br

Abstract. *This work is in the context of Computer Aided Design and its integration with cutting machines. An integration solution between drawings in CAD software built with the cut made by a Computerized Numerical Control (CNC) machine was developed. This integration was made by analyzing and converting the text file produced by CAD, extended to a file in XML format with filtering of relevant data and sent to a program (CAM) that simulates the behavior of a CNC cutting machine through an Arduino microcontroller.*

Resumo. *O presente trabalho encontra-se no contexto de Desenho Auxiliado por Computador e a sua integração com máquinas de corte. Foi desenvolvida uma solução de integração entre desenhos construídos em softwares CAD com o corte realizado por uma máquina Comando Numérico Computadorizado (CNC). Essa integração foi feita pela análise e conversão do arquivo texto produzido pelo CAD, estendido para um arquivo no formato XML com uma filtragem de dados relevantes e encaminhado a um programa (CAM) que simulará o comportamento de uma máquina de corte CNC através de um microcontrolador Arduino.*

1. Introdução

Este trabalho está inserido na área de Sistemas de Informação, mais precisamente em sistemas que auxiliam no projeto e no desenvolvimento de produtos na área industrial. Esses sistemas são conhecidos como *Computer Aided Drafting/Design* (CAD), *Computer Aided Manufacturing* (CAM) e *Computer Numeric Control* (CNC).

Quanto aos sistemas CAD, Souza e Ulbrich (2009) descrevem a palavra *drafting* como desenho técnico, que é a utilização de vistas em duas dimensões (2D) para representar um objeto tridimensional (3D).

Os sistemas CAD podem possuir integração com os *softwares* CAM, um sistema que é empregado na criação de programas para operação de máquinas CNC. Os Comandos Numéricos Computadorizados são comandos ou instruções de controle de máquinas de corte, de fresamento, usinagem, entre outras, ou seja, qualquer máquina que trabalhe sobre eixos de coordenadas cartesianas [Souza Ulbrich 2009].

Neste trabalho, a máquina que foi simulada é conhecida como máquina de corte, que tem como objetivo realizar cortes em chapas metálicas, geralmente em indústrias metalúrgicas, na manufatura de componentes para equipamentos de pequeno à grande

porte. Essa máquina tem como principais componentes a mesa de trabalho, eixo-árvore, motores de acionamento e respectivos controladores [Tecnopampa 2013].

Dentro da área industrial, há no mercado uma baixa variedade de fornecedores de *softwares* CAM ou ferramentas que realizem a integração entre sistemas CAD (nível conceitual) e motores de corte (nível físico). Além disso, as soluções disponíveis têm alto custo associado, ou seja, essas soluções são escassas e com um alto valor embarcado para esse serviço de integração.

Para realizar a integração entre o desenho feito em CAD com uma máquina de corte CNC foi realizado um estudo sobre os formatos de arquivos padrões gerados por estes *softwares*. A partir deste trabalho, verificou-se a existência de dois formatos de arquivos CAD universais o *Drawing* (DWG) e o *Drawing Interchange File Format* (DXF) [Autodesk 2007].

O DXF foi o formato selecionado para executar o projeto, pois é um arquivo textual com forma de escrita similar ao padrão *Extensible Markup Language* (XML), o que facilita a interpretação e manipulação do mesmo.

A partir da interpretação de um desenho no formato DXF, foi criado um novo arquivo no padrão de escrita XML com a API (*Application Programming Interface*) JDOM (*Java Document Object Model*), contendo informações de corte para uma máquina CNC. Esta interpretação foi realizada por um *software* CAM que abstrai as informações irrelevantes para o processo de manufatura da peça e gera um novo arquivo padrão de corte. O novo arquivo contém informações para a manufatura da peça, tais como coordenadas de corte, especificação do material e espessura. Os dados contidos no arquivo são processados pelo *software* CAM, que envia os comandos de corte para um microcontrolador Arduino, simulando o controle dos motores de uma máquina de corte CNC.

Para auxiliar no entendimento do artigo, o texto está dividido em 6 seções. A próxima seção – Referencial Teórico – trata de conceitos, tecnologias relacionadas e também de obras correlatas que ajudaram no desenvolvimento deste trabalho. A seção 3 mostra a metodologia escolhida, ferramentas utilizadas e a proposta do trabalho. A seção 4 mostra o desenvolvimento do trabalho. A seção 5 apresenta os resultados alcançados, dificuldades e trabalhos futuros. A última contém todas as referências utilizadas na elaboração deste estudo.

2. Referencial Teórico

A seção que segue trata de conceitos, tecnologias relacionadas e também de obras correlatas.

2.1. Desenho Técnico e Ferramentas de Apoio

Nesta seção são abordadas as definições de desenhos técnicos e ferramentas utilizadas para o desenvolvimento de desenhos e projetos.

A importância do desenho técnico se dá devido à necessidade que o homem teve, desde sempre, de se comunicar com seu semelhante. Sendo o Desenho Técnico uma linguagem, como tal deve ter uma gramática, uma ortografia e uma caligrafia própria, cujo estudo é necessário a quem pretenda ler e escrever corretamente essa linguagem [Veiga 1999].

O desenho técnico é a representação de um objeto destinado a explicar com rigor suas formas e dimensões, bem como os aspectos relevantes, para a sua produção. O desenho técnico deve ser perfeitamente perceptível e sem ambiguidades na forma como descreve um determinado objeto [Silva 2006]. É o método utilizado para representar objetos tridimensionais em um plano (folha de papel). Utiliza-se a representação por diferentes vistas do objeto, permitindo representá-lo em uma folha de desenho em duas dimensões [Souza 2009].

De acordo com Silva (2006), um sistema CAD é a expressão comercial para um conjunto de programas de computação gráfica hierarquizados segundo uma estrutura lógica de utilização, podendo ou não ser específicos, para obtenção de um dado tipo de desenho ou modelo inerente a um determinado domínio de aplicação (Engenharia Civil, Arquitetura, Engenharia Mecânica, entre outras).

A Figura 1 exemplifica um desenho de uma peça metálica feita em CAD, mostrando as projeções do componente e uma vista isométrica para melhor entendimento. Este desenho contém todas as informações necessárias para a fabricação do componente, como dimensões e material a ser utilizado.

Devido à grande diversidade de *softwares* CAD encontrados no mercado, surgiu a necessidade de padronização do formato de arquivo gerado por estes *softwares*, como o padrão de formato de arquivo *Drawing Interchange File Format* (DXF) ou Formato de Intercâmbio de Arquivos, e o formato *Drawing* (DWG) ou formato Desenho, sendo estes formatos os mais utilizados [Autodesk 2007].

O formato DXF tem uma peculiar forma de escrita que se assemelha com a organização por etiquetas e *tags* dos arquivos *Extensible Markup Language* (XML).

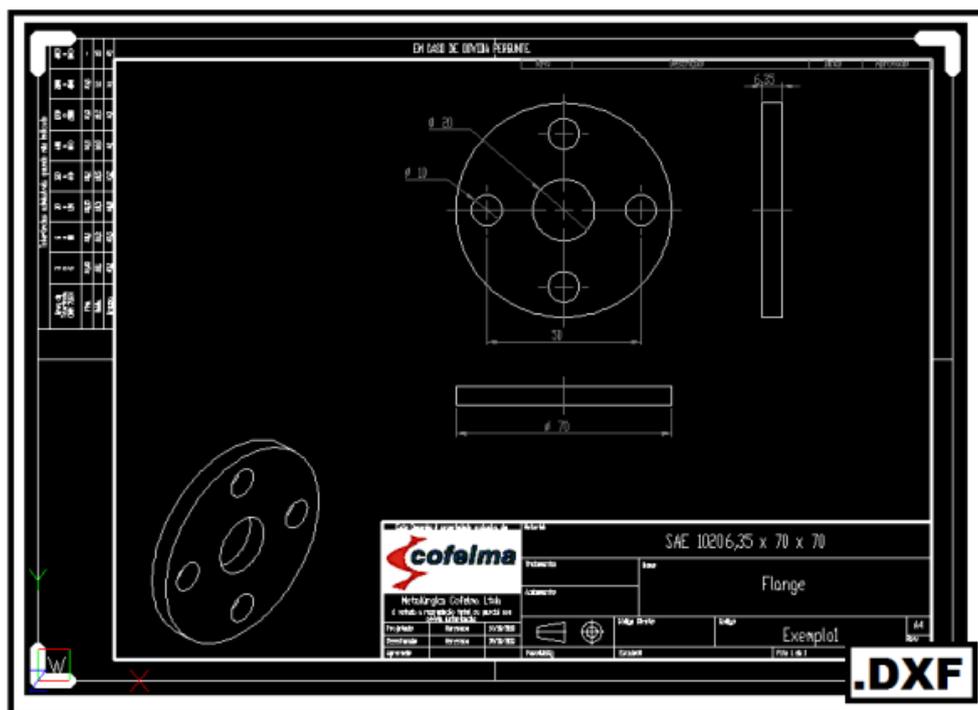


Figura 1 - Desenho 2D em CAD

2.2. Forma de Organização dos Arquivos XML E DXF

Nesta seção é abordada a finalidade do formato de arquivo *Drawing Interchange File Format* (DXF) para este trabalho, sendo mostradas as suas características e a forma básica de organização deste formato. Também é explanado sobre o formato de arquivo *Extensible Markup Language* (XML), que serão utilizados para padronização no setup ou configuração da máquina CNC.

A linguagem XML foi proposta pelo W3C (*World Wide Web consortium*) como uma nova alternativa à linguagem HTML (*Hypertext Markup Language*), devido ao limite de *tags* (forma de marcação de documento com início e fim) imposto pela mesma [Filho, 2004].

A *Extensible Markup Language* é um formato simples, baseado em texto para representar informações estruturadas: documentos, dados de configuração, livros, transações, faturas e muito mais. Foi criado a partir de um formato padrão mais antigo denominado SGML (ISO 8879) (*Standard Generalized Markup Language*), a fim de ser mais apropriado para o uso da Web [W3C 2010].

Para a construção de um documento XML, utilizam-se dois conceitos básicos de estruturação: os elementos e os atributos. Os atributos são utilizados como linguagem de descrição do documento como HTML e SGML, oferecendo informações adicionais que descrevem os elementos. Os elementos são divididos em complexos que são construídos com base em outros elementos hierarquicamente, e os elementos simples que contêm valores e dados [Navathe 2011].

Quanto à padronização de escrita de documentos XML, para facilitar o processamento do arquivo, tem-se um conjunto de funções de API chamado JDOM, que

permite que os programas manipulem a representação de árvore resultante correspondente a um documento XML bem formatado [JDOM, 2014].

O formato de arquivo DXF é uma representação de todas as informações contidas em desenhos CAD, com organização similar a do arquivo no formato XML, sendo este criado por AutoCAD (Empresa desenvolvedora de *softwares* CAD). Cada elemento do arquivo é percebido por uma *tag*, representado por um número inteiro que é chamado de código do grupo (Figura 2). O valor neste código de grupo indica o tipo de dados que o elemento contém, e também o significado do tipo de dado para um determinado objeto (registro). Praticamente toda a informação em um arquivo de desenho CAD pode ser representada pelo formato DXF [Autocad 2007].

LINE

The following group codes apply to line entities. In addition to the group codes described here, see Common Group Codes for Entities (page 75). For information about abbreviations and formatting used in this table, see Formatting Conventions in This Reference (page 2).

Line group codes

Group code	Description
100	Subclass marker (AcDbLine)
39	Thickness (optional; default = 0)
10	Start point (in WCS) DXF: X value; APP: 3D point
20, 30	DXF: Y and Z values of start point (in WCS)
11	Endpoint (in WCS) DXF: X value; APP: 3D point
21, 31	DXF: Y and Z values of endpoint (in WCS)
210	Extrusion direction (optional; default = 0, 0, 1) DXF: X value; APP: 3D vector
220, 230	DXF: Y and Z values of extrusion direction (optional)

Figura 2 - Formato de um código de grupo em arquivos DXF [Autoesk 2007].

2.3. CAM, Microcontroladores e CNC

Nesta seção do projeto, são abordadas as tecnologias CAM, CNC e microcontroladores, mostrando qual o *hardware* de prototipação foi escolhido para a realização deste trabalho. Também são abordadas as características e conceitos de máquina CAM, CNC e microcontroladores.

Para conseguir uma maior agilidade e padronização na manufatura de produtos, fez-se necessário criar uma interligação entre máquinas CNC (*Computer Numerical Control*) e os projetos das peças a serem fabricadas. Para este fim, foram desenvolvidos

os *softwares* CAM (*Computer Aided Manufacturing*) que são responsáveis pela geração de comandos para as máquinas a partir de desenhos CAD [Souza Ulbrich 2009].

A tecnologia CAM é um sistema de computação utilizado para planejar, gerenciar e controlar as operações de fabricação, estando ligado diretamente com os recursos de produção, como o desenho a ser materializado [Xu He 2004].

Um sistema CAM pode ser empregado na geração de programas CNC com função de realizar corte de materiais, torneamento, furação, retificação entre outros processos. Este sistema trabalha realizando cálculos de trajetória da ferramenta da máquina CNC com base em um desenho feito em CAD, fazendo com que o item fabricado tenha a mesma forma e medidas do desenho [Souza Ulbrich 2009].

Para realizar a comunicação de computadores com máquinas CNC, podem ser utilizados microcontroladores, que segundo Souza (2003) é como um “pequeno” componente eletrônico, dotado de uma “inteligência” programável, utilizados para controles de processos lógicos. Sendo todo o microcontrolador programável, pois toda a lógica de operação é estruturada na forma de um programa e gravada dentro do componente.

Devido à praticidade e baixo custo, para este projeto, é utilizado o microcontrolador Arduino, que é uma plataforma de prototipagem flexível *open-source*, de fácil uso de *software* e *hardware*.

2.4. Trabalhos Relacionados

Nesta seção são mostrados alguns trabalhos correlatos que serviram como base para o desenvolvimento deste projeto de integração.

2.4.1. CAD/CAM *solutions for STEP-compliant CNC manufacture*

Este projeto realizado por Newman e Rosso (2005) mostra que as tecnologias utilizadas para gerar o caminho de uma ferramenta NC (*Numerically Controlled*) ainda são baseadas em padrões definidos em 1950. O artigo fornece uma visão futurista de como este novo padrão pode ser utilizado na fabricação e destaca a evolução dentro dos sistemas CAD/CAM. Por fim, este artigo propõe um sistema compatível com CAD/CAM STEP.

2.4.2. *Striving for a total integration of CAD, CAPP, CAM and CNC*

O trabalho realizado por Xu e He (2004) mostra um comparativo entre programação realizada nos últimos 50 anos em cima da norma ISO 6983, ou também conhecida como códigos G, em relação à nova proposta de padrão da ISSO 10303 (STEP). Esta pesquisa foi realizada ao perceber uma série de impedimentos para realização de integração total CAD/CAM ao se utilizar a ISO 6983.

2.4.3. *Closed-Loop CAPP/CAM/CNC process chain based on STEP and STEP-NC inspection tasks*

Brecher, Vitr e Wolf (2010) descrevem neste trabalho o projeto da ISO 14649 Parte 16, o qual estipula os dados para contato de sondagem de inspeção, que permite a integração de tarefas de inspeção em uma sequência de operação de usinagem. É feito um breve

resumo de como Parte 16 se encaixa em outras normas de inspeção, intercâmbio de dados e projetos de norma, tais como DMIS (*Dimensional Measuring Interfaces Standards*) e DML (*Dimensional Markup Language*).

3. Metodologia, Ferramentas e Proposta

Nesta seção, aborda-se a metodologia escolhida, ferramentas utilizadas e a proposta do trabalho.

3.1. Proposta

Neste trabalho, foi padronizado um arquivo, no formato XML, contendo informações de corte para uma máquina CNC. Esta padronização é realizada através da simulação do corte de uma peça a partir da leitura do seu desenho 2D CAD no formato DXF. Esta leitura é realizada por um *software* CAM que abstrai informações irrelevantes para o corte e gera um novo arquivo padrão de corte, contendo informações para a manufatura da peça, tais como coordenadas de corte, material e espessura da peça. As informações contidas no arquivo são interpretadas pelo *software* CAM que comandará a máquina CNC através do microcontrolador Arduino. A máquina em questão teve seus motores representados por LED's (*Light Emitting Diode*) As Figura 3 e Figura 4 auxiliam no entendimento.

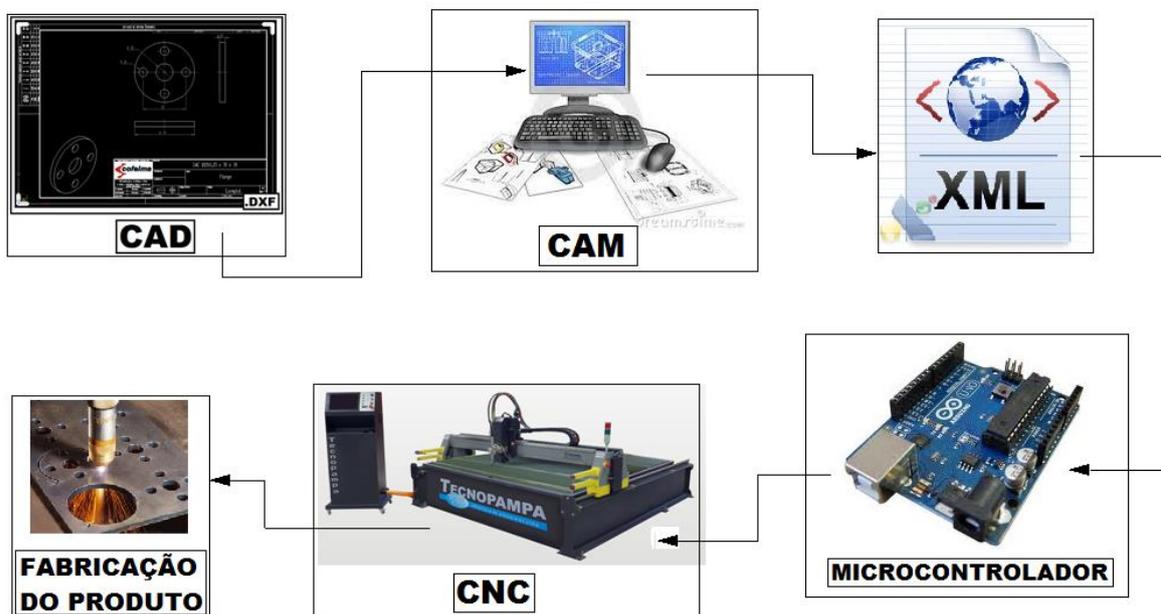


Figura 3 - Proposta de integração CAD, CAM e CNC.

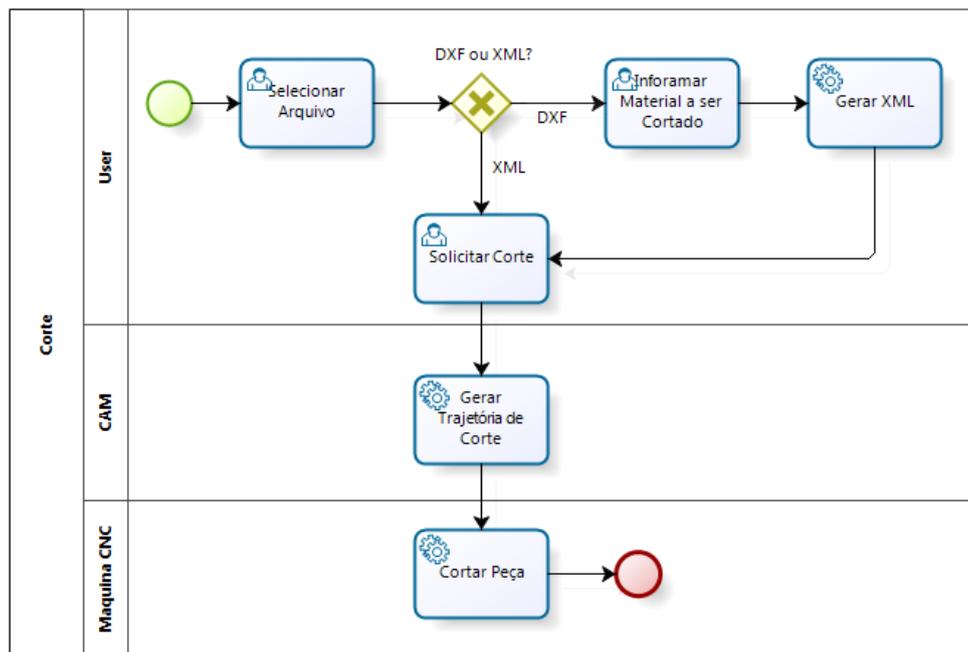


Figura 4 - Diagrama de atividade da integração CAD, CAM e CNC.

3. 2. Metodologia

A metodologia escolhida para ser utilizada neste projeto é a *Feature Driven Development* – Desenvolvimento Guiado por Características (FDD). Segundo Pressman (2005), é uma metodologia ágil de desenvolvimento que encoraja a satisfação do cliente, entrega incremental do *software*, equipes de projetos pequenas altamente motivadas, métodos informais, produtos de trabalhos de engenharia de *software* mínimos e simplicidade global no desenvolvimento.

3. 3. Ferramentas

As ferramentas utilizadas para o desenvolvimento deste projeto foram:

- NetBeans [Netbeans 2014] : IDE utilizada para desenvolver o *software* em linguagem Java;
- JDOM [JDOM, 2014]: API usado na geração e interpretação dos arquivos XML;
- *RXTX Java library* [RXTX 2014]: biblioteca para comunicação com o microcontrolador Arduino;
- *Software Arduino* [Arduino 2014]: aplicado para a programação do microcontrolador;
- Draftsight [3DS 2014]: programa gratuito da 3DS (*Dassault Systemes*), empregado para gerar os desenhos CAD 2D;
- Fritzing [Fritzing 2014]: *software open-source* que foi aplicado na modelagem do *hardware*.
- Bizagi [Bizagi 2014]: ferramenta escolhida para a modelagem dos digramas.

4. Resultados

Para a obtenção de um resultado “visível”, os motores da máquina de corte foram representados pela ligação entre um microcontrolador Arduino e LED’s. Observando a Figura 5, que representa o *hardware* utilizado neste projeto, pode-se perceber que são utilizados LED’s de duas cores, o azul representando o motor que se desloca no eixo cartesiano “x” e o vermelho representando o motor que se desloca no eixo cartesiano “y”.

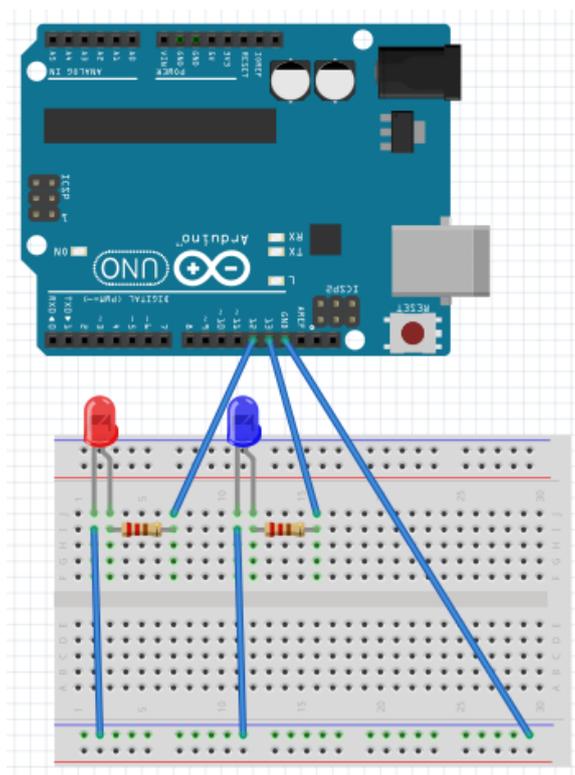


Figura 5 - *Hardware* de simulação da máquina de corte

A programação do microcontrolador é mostrada pela Figura 6. Ela trata o comando recebido pela porta serial: se o comando for igual a “1”, o Arduino liga o LED azul através do pino de saída 13; se for “2”, desliga o LED azul também através do pino de saída 13; se for “3”, liga o LED vermelho pelo pino de saída 14; por fim, se for “4”, desliga o LED vermelho pelo pino de saída 14. Para melhor entendimento da ligação dos pinos, observar a Figura 5 que ilustra a *hardware*.

```

void loop () {
  if (Serial.available() > 0) {
    byteEntrada = Serial.read();
    if (byteEntrada == '1') {
      digitalWrite(LED AZUL, LOW);
    }
    else if (byteEntrada == '2') {
      digitalWrite(LED AZUL, HIGH);
    }
    if (byteEntrada == '3') {
      digitalWrite(LED VERMELHO, LOW);
    }
    else if (byteEntrada == '4') {
      digitalWrite(LED VERMELHO, HIGH);
    }
  }
}
}

```

Figura 6 - Programação do microcontrolador

Para conseguir realizar o corte de uma peça é necessário importar o arquivo para o *software* CAM através da opção “arquivo/abrir” (Figura 7). Quando um arquivo no formato DXF ou XML é importado, o programa gera um objeto com base no arquivo e o exibe na tela (Figura 8).

No *software* CAM foram implementadas as classes e funções relativas a cortes de polígonos, ou seja, o programa interpreta e corta somente peças formadas por pontos e linhas com uma forma fechada (todos os pontos conectados). Podem ser desenvolvidas outras classes e métodos para corte de outros formatos (círculos, elipses *spline*), dependendo da finalidade ou aplicação do projeto.

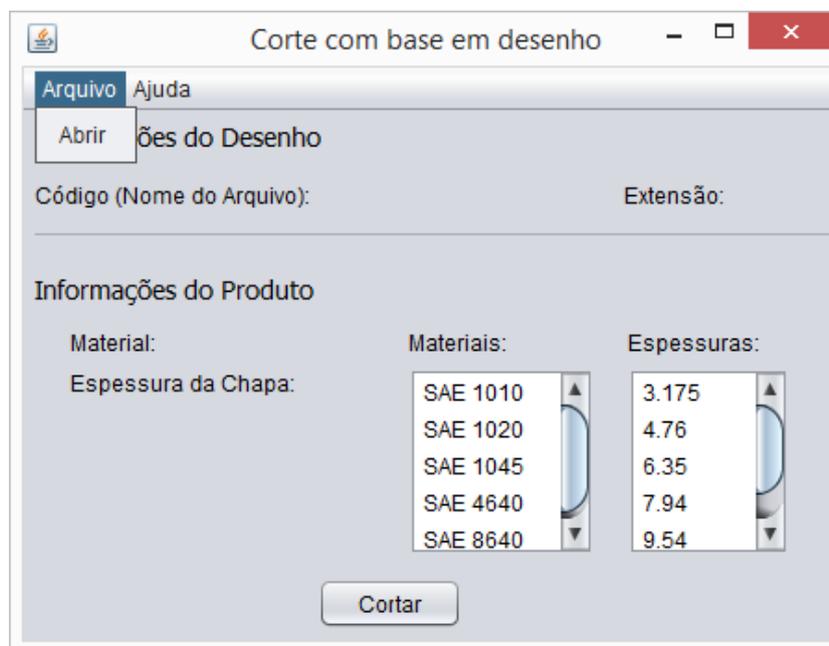


Figura 7 - Tela Principal Software CAM

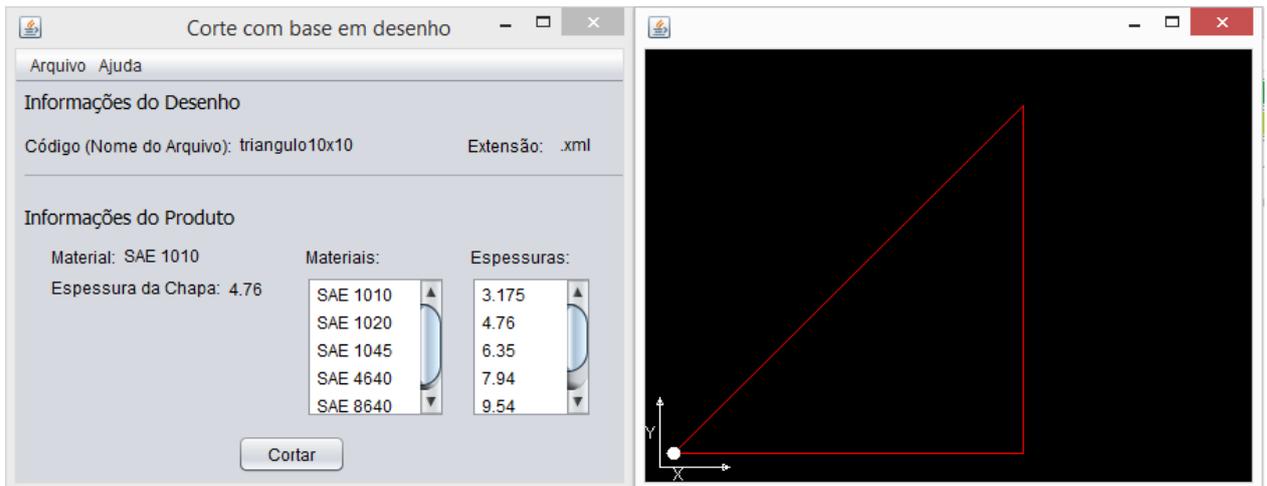


Figura 8 - CAM com Tela de Visualização do Item a ser Cortado

Para realizar o corte do arquivo exibido na tela é necessário selecionar a opção “cortar” mostrada na Figura 8. O programa calcula a trajetória de corte com base no objeto gerado pela interpretação do arquivo XML ou DXF.

Para enviar dados de corte da peça, o *software* CAM calcula a trajetória de corte através da inserção de pontos na função da reta, onde $y=ax+b$. Na função citada, “y” representa o valor do ponto no eixo cartesiano y, “x” representa o valor do ponto no eixo cartesiano x, “a” representa o coeficiente angular e o “b” coeficiente linear.

Para melhor entendimento, pode-se observar a Figura 9 que mostra o trecho de código para corte de uma linha vertical. Esta parte do código verifica se a variação de valores do eixo “x” é igual à zero, sendo esta condição verdadeira somente o eixo “y” sofrerá variação. Logo, será ligado somente o motor representado pelo LED vermelho que se desloca no eixo “y”. Este motor terá um deslocamento igual à variação de “y” representada por “deltaY”.

```

//reta vertical
if (deltaX == 0) {
    int j = 0;
    while (j < abs(deltaY)) {
        j++;
        //atualizaTela
        if (deltaY > 0) {
            this.panel.atualizaPontoDeCorte(vetX[i], vetY[i] + j);
        } else {
            this.panel.atualizaPontoDeCorte(vetX[i], vetY[i] - j);
        }
        //aciona motor
        enviarDados(vermelhoON);
        sleep(timeOn);
        enviarDados(vermelhoOFF);
        sleep(timeOff);
    }
}
}

```

Figura 9 - Código para o calculo de trajetória de uma linha vertical

Ao selecionar a opção de “cortar”, se o arquivo importado for do tipo DXF o *software* automaticamente gera o arquivo XML correspondente a peça. Além disso, acrescenta as informações de especificação de material e espessura, que são inseridas pelo usuário (Figura 11).

A classe XML do *software* cria a estrutura de um polígono através da API JDOM, criando os elementos e os organizando dentro da estrutura do arquivo (Figura 10).

```

//Cria os elementos do xml
Element desenho = new Element("desenho");
Element poligono = new Element("poligono");
Element valoresX = new Element("valoresX");
Element valoresY = new Element("valoresY");
Element qtdPontos = new Element("qtdPontos");
Element espessuraMat = new Element("espessura");
Element especificacaoMat = new Element("especificacao");
//Organiza a estrutura do xml
poligono.addContent(valoresX);
poligono.addContent(valoresY);
poligono.addContent(qtdPontos);
desenho.setAttribute("Código", nomeArquivo);
desenho.addContent(poligono);
desenho.addContent(espessuraMat);
desenho.addContent(especificacaoMat);

```

Figura 10 - Criação da Estrutura XML no JDOM

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <desenho Código="triangulo10x10">
3   <poligono>
4     <valoresX>0.0;10.0;10.0;</valoresX>
5     <valoresY>0.0;0.0;10.0;</valoresY>
6     <qtdPontos>3</qtdPontos>
7   </poligono>
8   <espessura>4.76</espessura>
9   <especificacao>SAE 1010</especificacao>
10 </desenho>
```

Figura 11 - Arquivo XML Gerado na Interpretação do desenho DXF de um Triângulo 10x10

```
7010      3
7011      91
7012      2
7013      300
7014      DATA
7015      309
7016      CELLSTYLE_END
7017      0
7018      ENDSEC
7019      0
7020      EOF — Final do Arquivo
```

Figura 12 - Final de um arquivo DXF do desenho de triângulo 10x10

Gerando este arquivo XML, no exemplo de um triângulo (Figura 11), reduz-se de um arquivo DXF com 7020 linhas (Figura 12), com muitas informações desnecessárias para o processo de corte, para um arquivo XML de 10 linhas contendo todas as informações necessárias.

5. Conclusões

Para entendimento dos resultados são abordados neste projeto desenhos técnicos, formas de organização de arquivos (XML e DXF), integração das tecnologias CAD, CAM e CNC ferramentas utilizadas e metodologia.

Com a interpretação de um desenho no formato DXF, foi criado um novo arquivo no padrão de escrita XML com o API JDOM, contendo informações de corte para uma máquina CNC. Esta interpretação foi realizada por um *software* CAM que abstrai as informações irrelevantes para o processo de manufatura da peça e gera um novo arquivo padrão de corte. O novo arquivo contém informações para a manufatura da peça, tais como coordenadas de corte, especificação do material e espessura. Os dados contidos no arquivo são processados pelo *software* CAM, que envia os comandos de corte para um microcontrolador Arduino, simulando o controle dos motores de uma máquina de corte CNC.

O estudo demonstrou a viabilidade e eficiência na utilização de um novo padrão de arquivo de corte XML, que contém todas as informações necessárias para o corte de uma peça, ainda possuindo um tamanho muito menor que os arquivos DXF gerados pelos desenhos em CAD. Esta viabilidade foi comprovada através da integração entre desenhos CAD, *software* CAM e máquina de corte CNC, representada através da utilização de um *hardware* comandado pelo microcontrolador Arduino.

No desenvolvimento do trabalho, as maiores dificuldades encontradas foram a pouca literatura disponível com temas correlatos, difícil interpretação matemática dos arquivos DXF e aplicação matemática para realização das rotas de corte.

Como trabalhos futuros, sugere-se desenvolver uma máquina de corte CNC apropriada com motores, criar novos objetos de corte para formas circulares, elípticas e *spilines* e medir a diferença de processamento para corte de um arquivo XML gerado em relação a um DXF.

6. Referências

- Arduino (2013) “Arduino *Software*”, <http://arduino.cc>, setembro.
- Autodesk, (2013) “DXF Reference. 2007”, http://images.autodesk.com/adsk/files/acad_dxf0.pdf, outubro.
- Bizagi (2014) “Bizagi Business Process Management(BPMS)”, <http://www.bizagi.com>, Fevereiro.
- Brecher, C.; Vitr; e Wolf, J. (2010) “International Journal of Computer Integrated Manufacturing. Closed-Loop CAPP/CAM/CNC process chain based on STEP and STEP-NC inspection tasks”, ScholarOne, p. 1-20.
- Filho, A. M. da S. (2004) “Programando com XML”, Elsevier, Rio de Janeiro.
- Fritzing (2014) “Fritzing”, <http://fritzing.org/home>, Março.
- JDOM (2014) “JDOM 2.0.5”, <http://www.jdom.org>, Fevereiro.
- Navathe, E. (2011) “Sistemas de banco de dados”, 6ª.ed. Pearson Addison Wesley, São Paulo.
- Netbeans (2014) “Netbeans IDE”, <http://netbeans.org>, Fevereiro.
- Newman, S. T.; Allen, R. D.; Rosso, Jr. R. S. U. (2003) “Computer Integrated Manufacturing. CAD/CAM solutions for SETP-compliant CNC manufacture” .Taylor & Francis Ltd. p. 590-597.
- Pressman, R. S. (2005) “Engenharia de *Software*”, 6ª ed. McGraw-Holl Interamericana do Brasil Ltda, São Paulo.
- RXTX (2014) “RXTX Java Library”, <http://www.playground.arduino.cc/interfacing/java>, Fevereiro.
- Silva A. (2006) “Desenho Técnico Moderno”, 4ª.ed. LTC, Rio de Janeiro.
- Souza, A. F. de; ULBRICH, C. B. L. (2009) “Engenharia integrada por computador e sistemas: CAD/CAM/CNC: Princípios e aplicações”, Artliber Editora, São Paulo.
- Souza, D. J. de. (2003) “Desbravando o PIC”, 6ª ed. Érica Ltda, São Paulo.

- Tecnopampa (2013) “Tecnopampa Industria de Maquinas LTDA”
<http://www.tecnopampa.com.br/#!tecnocut/c8tx>, setembro.
- Veiga, L. (1999) “Desenho Técnico”, 11ªed. Fundação CalousteGulbenkian, Lisboa
- W3C. (2013) “XML Essencials 2010”, <http://www.w3.org/standards/xml/core>, outubro.
- Xu, X. W.; He, Q. (2004) “Robotic and Computer-Integrated Manufacturing. Striving for a total integration of CAD, CAPP, CAM and CNC”, Elsevier Ltd. p. 101-109.
- 3DS (2014) “Dassaul Systemes”, <http://www.3ds.com>, Maio.