

**FACULDADE DE TECNOLOGIA DE SÃO BERNARDO DO CAMPO
“ADIB MOISÉS DIB”**

**EWERTON DA SILVA COELHO
FRANCISCO EDMILSON PESSOA DE LIMA
HUGO PEREIRA DA SILVA
MALQUIAS NUNES DA SILVA**

AUTOMAÇÃO DO PLANO DE MANUTENÇÃO DA MÁQUINA CNC LASER

São Bernardo do Campo - SP
Junho/2018

**EWERTON DA SILVA COELHO
FRANCISCO EDMILSON PESSOA DE LIMA
HUGO PEREIRA DA SILVA
MALQUIAS NUNES DA SILVA**

AUTOMAÇÃO DO PLANO DE MANUTENÇÃO DA MÁQUINA CNC LASER

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Bernardo do Campo “Adib Moises Dib” como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial.

Orientador: Prof. Me. Valter Espindola Thomaz

Coorientador: Prof. Dr. Delcinio Ricci

São Bernardo do Campo - SP
Junho/2018

**EWERTON DA SILVA COELHO
FRANCISCO EDMILSON PESSOA DE LIMA
HUGO PEREIRA DA SILVA
MALQUIAS NUNES DA SILVA**

AUTOMAÇÃO DO PLANO DE MANUTENÇÃO DA MÁQUINA CNC LASER

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Bernardo do Campo “Adib Moises Dib” como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial.

Trabalho de Conclusão de Curso apresentado em: 28/06/2018

Banca Examinadora:

Prof. Me. Valter Espindola Thomaz, FATEC SBC - Orientador

Prof. Esp. Gervasio das Neves Salvador, FATEC SBC - Avaliador

Prof. Esp. Marcelo George Griese, FATEC SBC - Avaliador

Dedicamos esse trabalho aos nossos amores, pais, esposas e filhos. Dedicamos também a todos os docentes que compartilharam seus conhecimentos e aos amigos da Fatec, que em alguns momentos contribuíram com ideias e experiências.

Agradecemos ao Prof. Me. Valter Espindola Thomaz por sua orientação, que em tantos momentos foi fundamental para encontrarmos a direção mais correta na construção do nosso projeto. Agradecemos por sua condução suave e ao mesmo tempo esclarecedora. Mostrando as diversas soluções e caminhos ao longo do projeto.

Agradecemos ao Prof. Dr. Delcinio Ricci por sua sabedoria e dedicação, que transformou nossas ideias em propostas claras e de fácil leitura.

Agradecemos à Fatec São Bernardo do Campo pelas ferramentas e espaço cedido para tornar possível o desenvolvimento de nosso projeto.

“Determinação, coragem e autoconfiança são fatores decisivos para o sucesso. Se estamos possuídos por uma inabalável determinação conseguiremos superá-los. Independentemente das circunstâncias, devemos ser sempre humildes, recatados e despidos de orgulho”.

Dalai Lama

RESUMO

Neste trabalho é criado uma aplicação de um sistema supervisor, automatizando as informações do plano de manutenção preventiva, analisando os principais componentes dos sistemas e extraíndo as ações recomendadas, gerando em forma de alertas o tipo de ação a ser executada conforme o tipo de falha alarmado, proporcionando a visualização dos históricos de falhas, alarmes, gráficos e controle de vibração integrando a comunicação conforme o funcionamento da máquina CNC Laser de 3 eixos desenvolvida, montada, ajustada e programada. A máquina é usada para simular o funcionamento dos desvios em um processo e o seu monitoramento e controle dos principais componentes se faz por meio de sensor de vibração ligado a placa de controle Arduino Mega2560. A automação do plano de manutenção no sistema supervisor permite a gestão e monitoramento da máquina em tempo real, proporcionando uma significativa redução nos custos operacionais, tempo de execução no diagnóstico de falhas, na padronização das ações recomendadas do plano de manutenção e maior aumento na disponibilidade da máquina.

Palavras-chaves: Supervisor. CNC. Laser. Sensor de Vibração. Automação

ABSTRACT

In this work is created an application of a supervisory system, automating the information of the preventive maintenance plan, analyzing the main components of the systems and extracting the recommended actions, generating in the form of alerts the type of action to be executed according to the type of alarm , providing visualization of fault histories, alarms, graphs and vibration control, integrating the communication according to the operation of the CNC laser machine developed, assembled, adjusted and programmed. The machine is used to simulate the operation of deviations in a process and its monitoring and control of the main components is done by means of vibration sensor connected to the control board Arduino Mega2560. The automation of the maintenance plan in the supervisory system allows the management and monitoring of the machine in real time, providing a significant reduction in operational costs, runtime in fault diagnosis, standardization of the recommended actions of the maintenance plan and greater increase in availability of the machine.

Keywords: Supervisory. CNC. Laser. Vibration Sensor. Automation

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1.1 – Máquina fresadora CNC controlada por cartão perfurado..... | 17 |
| Figura 1.2 – Funcionamento do CNC..... | 19 |
| Figura 1.3 – Exemplo de ponto referencial..... | 19 |
| Figura 1.4 – Coordenadas percorridas pela ferramenta..... | 20 |
| Figura 1.5 – Processamento do CNC..... | 21 |
| Figura 1.6 – Corte a Laser..... | 23 |
| Figura 1.7 – Placa Arduino Mega2560..... | 24 |
| Figura 1.8 – Alimentação da placa Arduino Mega2560..... | 25 |
| Figura 1.9 – Pinagem do sensor MPU-6050..... | 26 |
| Figura 1.10 – Diagrama de bloco de funcionamento do MPU-6050..... | 28 |
| Figura 1.11 – Funcionamento das condições START e STOP..... | 29 |
| Figura 1.12 – Interfaces do <i>design</i> da interação..... | 30 |
| Figura 1.13 – Sistema supervisorío..... | 31 |
| Figura 1.14 – Etapas da FMEA..... | 32 |
| Figura 2.1 – Diagrama de blocos do projeto..... | 36 |
| Figura 3.1 – Projeto finalizado..... | 40 |
| Figura 3.2 – Diagrama de funcionamento do projeto..... | 41 |
| Figura 3.3 – Estrutura do CNC montada..... | 42 |
| Figura 3.4 – Montagem da mesa do CNC..... | 43 |
| Figura 3.5 – Montagem do carro móvel do CNC..... | 44 |
| Figura 3.6a – Montagem finalizada do CNC..... | 45 |
| Figura 3.6b – Disposição dos componentes do CNC..... | 45 |
| Figura 3.7 – Layout da placa do controle do CNC..... | 46 |
| Figura 3.8 – Configuração da placa GRBL..... | 47 |
| Figura 3.9 – Configuração da placa controladora do CNC..... | 48 |
| Figura 3.10 – Movimentação dos eixos da máquina CNC..... | 49 |
| Figura 3.11 – Área de trabalho do supervisorío..... | 51 |
| Figura 3.12 – Tela Supervisão..... | 51 |
| Figura 3.13 – Tela Automação..... | 52 |
| Figura 3.14 – Programação da placa Arduino..... | 53 |
| Figura 3.15 – Sensor de vibração MPU-6050..... | 54 |

| | |
|--|----|
| Figura 3.16 – Ligação da placa Arduino Mega 2560 com o sensor MPU-6050 | 55 |
| Figura 3.17 – Montagem dos cabos de comunicação do sensor de vibração | 56 |
| Figura 3.18 – Arvore funcional da máquina CNC..... | 57 |
| Figura 3.19 – Funcionamento da FMEA..... | 58 |
| Figura 3.20 – Instalação do módulo laser na máquina CNC | 59 |
| Figura 3.21 – Interface do <i>software</i> LiteFire | 60 |
| Figura 3.22 – Impressão de imagem de teste | 61 |
| Figura 3.23 – Configuração da placa Arduino | 62 |
| Figura 3.24 – Configuração do supervisor..... | 63 |
| Figura 3.25 – Mensagem da ação recomendada | 65 |

LISTA DE TABELAS E GRÁFICOS

| | |
|---|----|
| Tabela 1.1 - Bloco de comando com letras | 22 |
| Tabela 2.1 – Materiais para montagem do projeto | 37 |
| Tabela 2.2 – Cronograma do desenvolvimento do projeto | 38 |
| Tabela 3.1 – Principais comandos de configuração da placa GRBL..... | 47 |

LISTA DE ABREVIATURAS

| | |
|-------|--|
| 2D | Duas Dimensões |
| 3D | Três Dimensões |
| A | Ampere |
| CN | Controle Numérico |
| CNC | Controle Numérico Computadorizado |
| DDE | <i>Dynamic Data Exchange</i> |
| FAA | Força Aérea Americana |
| FMEA | <i>Failure Mode and Effect Analysis</i> |
| GPS | <i>Global Positioning System</i> |
| IDE | <i>Integrated Development Environment</i> |
| IHM | Interface Homem-Máquina |
| IOT | <i>Internet of things</i> |
| Laser | <i>Light amplification by stimulated emission of radiation</i> |
| Log | Registro de eventos |
| MHz | Mega Hertz |
| MIT | <i>Massachusetts Institute of Technology</i> |
| mm | milímetros |
| Nm | Newton metro |
| PWM | <i>Pulse Width Modulation</i> |
| RPN | Coeficiente da Prioridade do Risco |
| SCADA | <i>Supervisory Control and Data Acquisition</i> |
| TAG | Palavra chave |
| USB | <i>Universal Serial Bus</i> |
| V | Volt |

SUMÁRIO

| | | |
|----------------------------|---|-----------|
| INTRODUÇÃO | 14 | |
| 1 | FUNDAMENTAÇÃO TEÓRICA | 17 |
| 1.1 | Controles numéricos computadorizados..... | 17 |
| 1.2 | Funcionamento e programação da máquina CNC..... | 18 |
| 1.3 | Tecnologia Laser | 22 |
| 1.4 | Arduino Mega2560..... | 24 |
| 1.5 | Sensor MPU-6050 | 26 |
| 1.6 | <i>Design</i> de interação e sistemas supervisórios | 29 |
| 1.7 | Plano de manutenção – FMEA..... | 31 |
| 2 | METODOLOGIA..... | 34 |
| 2.1 | Tema-problema com justificativa e descrição funcional do projeto | 35 |
| 2.2 | Etapas teóricas e práticas para o desenvolvimento do projeto | 37 |
| 3 | DESENVOLVIMENTO DO PROJETO | 40 |
| 3.1 | Montagem mecânica da máquina, ajustes e alinhamento | 42 |
| 3.2 | Instalação e configuração da placa de controle do CNC | 46 |
| 3.3 | Teste preliminar de funcionamento da máquina CNC..... | 48 |
| 3.4 | Desenvolvimento do supervisório com o <i>software</i> Elipse SCADA..... | 50 |
| 3.5 | Configuração da placa Arduino controladora do supervisório | 52 |
| 3.6 | Comunicação do sensor de vibração com o supervisório | 53 |
| 3.7 | Desenvolvimento e integração do plano de manutenção..... | 57 |
| 3.8 | Testes finais e funcionamento do projeto | 57 |
| 3.9 | Obstáculos e soluções durante o desenvolvimento do projeto..... | 65 |
| CONSIDERAÇÕES FINAIS | 67 | |
| REFERÊNCIAS..... | 69 | |

APÊNDICES71

ANEXOS72

INTRODUÇÃO

A manutenção preventiva dentro dos sistemas industriais permite uma nova realidade produtiva com relação à otimização, disponibilidade e confiabilidade das máquinas e equipamentos. Quando a manutenção preventiva é utilizada em conjunto aos mecanismos de supervisão, é possível obter as informações de produção, custo e segurança em tempo real, determinando previamente a necessidade de intervenções, reduzindo as desmontagens das máquinas para inspeções e manutenções corretivas, utilizando aparelhos que identificam mudanças na vibração, temperatura, pressão, nível de óleo e análise ferrográfica. Tornando possível prever folgas, desgastes prematuros e deterioração dos componentes antes que ocorram as falhas, através de intervenções que passaram a ser planejadas.

A gestão da manutenção faz o uso de diversas ferramentas de qualidade para garantir a maior eficiência e eficácia nas suas ações, visando ter um diagnóstico preciso, e com isso, maximizar a vida útil das máquinas e entender o ponto de falha de cada componente para reduzir o número de anomalias.

A análise das falhas de uma máquina serve para identificar os potenciais riscos de anomalias e também para criar um plano de manutenção preventiva, baseado nas informações de manuais, desenhos técnicos e na experiência do desenvolvedor do plano de manutenção, levantando informações técnicas dos componentes significativos e detalhando as devidas ações preventivas para padronizar um método de identificação de defeitos.

Acontece que todo esse trabalho é manual, a geração dos dados e o acompanhamento das frequências das rotinas criadas por esse plano dependem muito da mão de obra de um planejador, para que se possam distribuir todos esses documentos as equipes de manutenção, que fazem as atividades programadas retroalimentando um *software* para atualizar a situação da condição da atividade. Isso gera uma grande perda de tempo e gastos excessivos para manter o controle da periodicidade de cada tarefa.

Esse controle impacta diretamente o fluxo financeiro da empresa, diminuindo de forma drástica os efeitos colaterais que a má gestão da manutenção pode gerar. O custo excessivo com manutenções corretivas, tais como, custo/tempo da parada de linha e ineficiência da manutenção.

Das pontuações mencionadas, o objetivo desse trabalho que se intitula Automação do plano de manutenção da máquina CNC laser é montar uma máquina CNC de corte a Laser para criar um plano de manutenção preventiva voltado a ela. Justifica-se o uso da análise do funcionamento dos componentes por economia do maquinário, maior acessibilidade de manutenção, facilidade na interação homem-máquina (IHM), melhoria na supervisão de falhas através de telas de supervisórios, permissão do controle local das atividades em tempo real, minimização de custo, maior controle de manutenção, menor geração de resíduos e conseqüentemente uma redução no impacto ambiental.

Para o desenvolvimento do projeto, utiliza-se a máquina de comando numérico computadorizado (CNC), a IHM para supervisão, o Laser para corte e gravura de peças, o Arduino para o controle da máquina e do módulo laser, o sensor acelerômetro e um banco de dados para documentar as informações do plano de manutenção com o uso da ferramenta FMEA.

Este trabalho é constituído por três capítulos:

Capítulo 1 - Fundamentação Teórica: abordagem das teorias e conceitos de autores renomados que dão sustentação ao desenvolvimento e construção do projeto intitulado Automação do plano de manutenção da máquina CNC laser.

Capítulo 2 - Metodologia: encontram-se os métodos e técnicas que direcionam a trajetória para a construção e desenvolvimento do projeto.

Capítulo 3 - Desenvolvimento do Projeto: descreve passo a passo a construção do projeto, ilustrando-o com figuras e tabelas. Mostra a ligação entre a teoria pesquisada com o processo de montagem e funcionamento do projeto.

E finalmente, as Considerações Finais: faz-se uma retomada, em linhas gerais, do que é abordado desde o objetivo e a justificativa citados na introdução até a concretização da construção do projeto. Aponta as relações entre os fatos verificados e a teoria, as conquistas alcançadas, as vantagens e desvantagens e possíveis sugestões para estudos futuros.

1 FUNDAMENTAÇÃO TEÓRICA

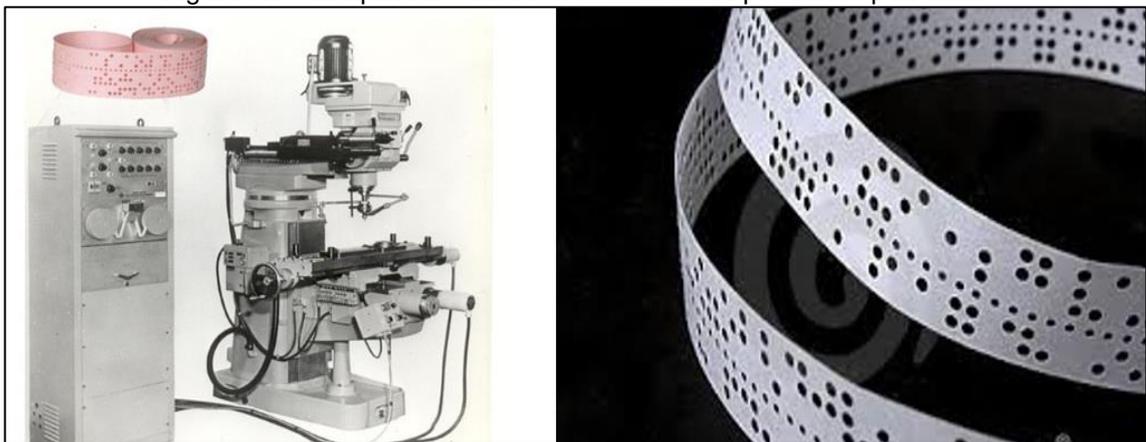
Neste capítulo encontram-se as teorias de autores renomados, que dão sustentação ao desenvolvimento e construção do projeto que se intitula Automação do plano de manutenção da máquina CNC laser. Essas teorias estão relacionadas à máquina CNC laser, a eletrônica relacionada à plataforma Arduino, e a automatização dos dados relacionados as ações de um plano de manutenção.

1.1 Controles numéricos computadorizados

Para Fitzpatrick (2013), o controle numérico computadorizado (CNC) é um dispositivo eletrônico capaz de receber informações e enviar sinais à máquina, de forma a executar operações sem a intervenção do operador. Essa tecnologia de usinagem surgiu de um protótipo da Parsons Corporation, que em 1947 testou em uma máquina manual o controle numérico (CN) baseado em cartões perfurados.

Nessa época o mundo vivia em uma instabilidade pós-guerra, e a Força Aérea Americana (FAA) constatou que o modo de produção manual dos componentes dos aviões era ineficaz em caso de emergência. A FAA investiu no desenvolvimento do projeto CNC da Parsons Corporation, que na época era conhecido por CN, a Figura 1.1 ilustra uma máquina fresadora CN controlada por cartão perfurado.

Figura 1.1 – Máquina fresadora CNC controlada por cartão perfurado



Fonte: www.proptimus.com.br, 2017

Fitzpatrick (2013) esclarece ainda que em 1953, no Instituto de Tecnologia de Massachusetts (MIT), foi demonstrado um protótipo CNC de uma máquina da Cincinnati. A partir de então, com o desenvolvimento tecnológico dos computadores, o CNC vem ganhando espaço em diversas aplicações industriais que revolucionaram os meios de produção. Tais como, os tornos, as fresadoras, as impressoras em três dimensões (3D) e CNC com corte à Laser.

Cassaniga (2005), alega que uma máquina que opera através de controle numérico computadorizado é a melhor opção para solucionar os problemas mais complexos na área de usinagem. Em situações que exigiam máquinas ou ferramentas específicas para se realizar determinadas operações, com o advento do CNC os projetos de construção de peças e ferramentas se realizam de formas bem mais simples.

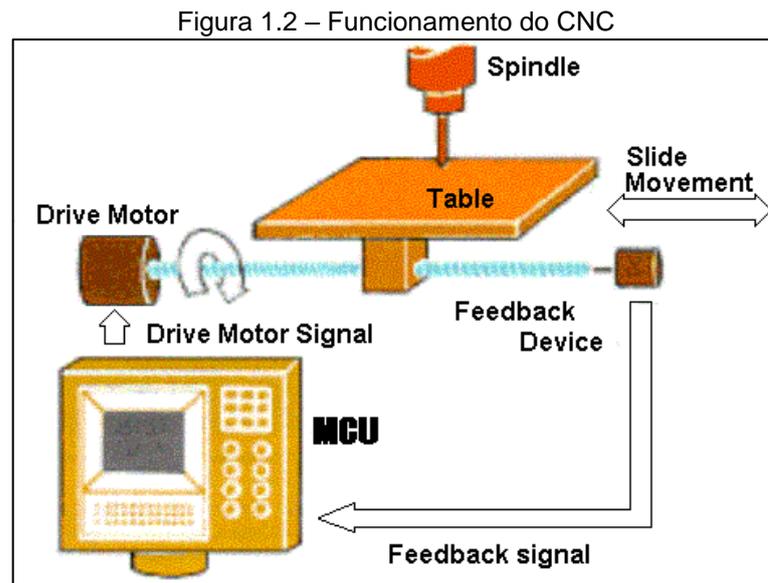
1.2 Funcionamento e programação da máquina CNC

Espinola e Pessoa (2014), informam que o CNC é uma máquina que realiza a sequência do seu trabalho mediante recurso programado, baseado em comandos numéricos, que é um sistema de coordenadas para elaboração de qualquer perfil geométrico x, y e z.

O sistema de coordenadas é formado por três eixos ortogonais, cujas direções obedecem à regra da mão direita. Em consequência dessa característica desenvolve-se uma máquina com grande precisão, poder de repetibilidade e com o mínimo de intervenção humana para execução de um trabalho. O princípio de funcionamento de um CNC, por questões de segurança, ocorre em circuitos fechados, utilizando sensores de posição para controle da máquina.

Os passos de funcionamento de uma máquina CNC, se desenvolvem após um comando enviado, de modo geral, por um computador, determinando quantas vezes, precisamente, o servo motor deve girar. O servo motor movimenta o fuso que se encontra num suporte da mesa e está instalado sobre o barramento da máquina.

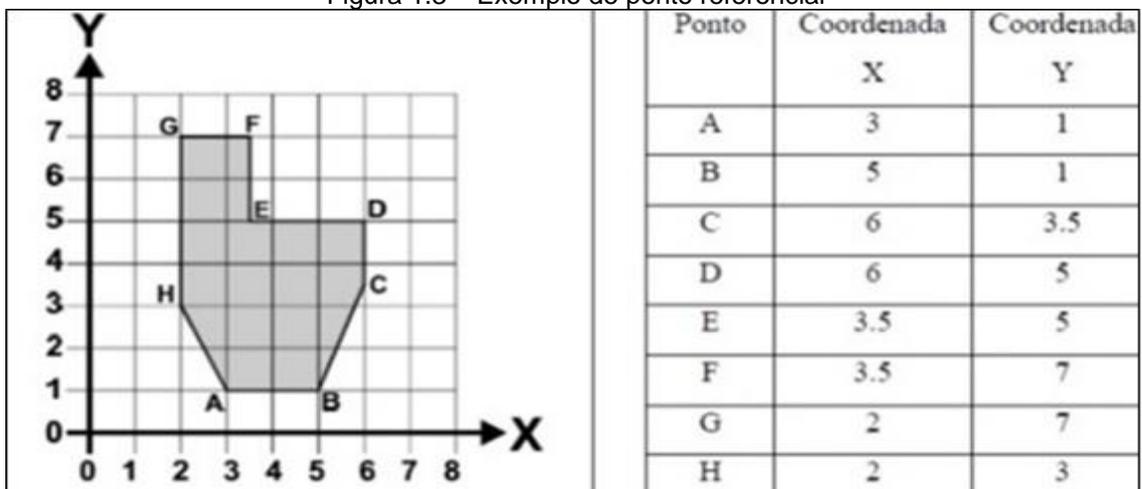
Quando o fuso gira os passos do mesmo, força a mesa em uma direção, movimentando o eixo linear. Um módulo comparador anexado ao fuso proporciona o eficaz controle dimensional, verificando se o número comandado de rotações foi atingido, a Figura 1.2 ilustra o funcionamento de uma máquina de controle numérico comandada por computador.



Fonte: www.mundodocnc.com.br, 2018

Espinola e Pessoa (2014) destacam a dificuldade de manipular a quantidade de rotação do servo motor para executar uma tarefa. A programação do CNC é baseada em coordenadas cartesianas/retangulares ou coordenadas polares. O programador determina na área a ser usinada o ponto zero, que é o ponto de referência para o percurso da ferramenta, como pode ser visto na Figura 1.3.

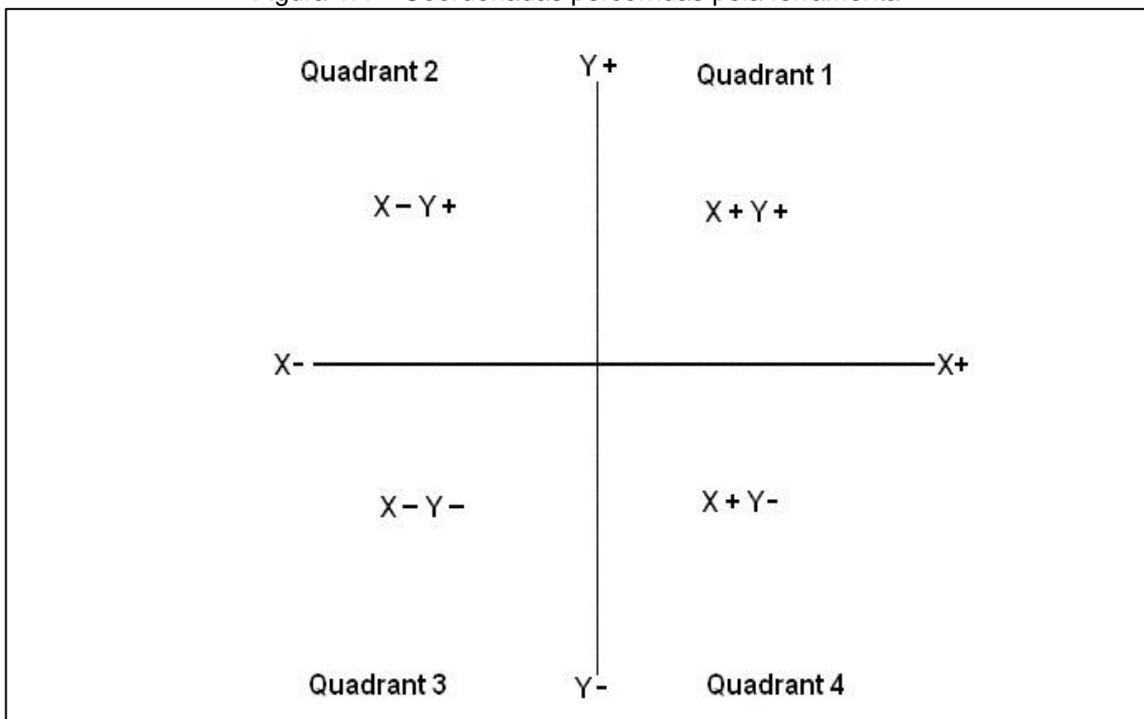
Figura 1.3 – Exemplo de ponto referencial



Fonte: www.starnc.com.br, 2017

Espinola e Pessoa (2014) informam que para executar o percurso, o programador determina os pontos no sistema de eixos X, Y e estabelece os pontos que devem ser percorridos pela ferramenta. Nesse caso, o quadrante percorrido é o superior da direita (quadrante 1); mas, pode ser o da direita inferior (quadrante 4), o da esquerda superior (quadrante 2) ou o da esquerda inferior (quadrante 3), conforme mostra a Figura 1.4.

Figura 1.4 – Coordenadas percorridas pela ferramenta



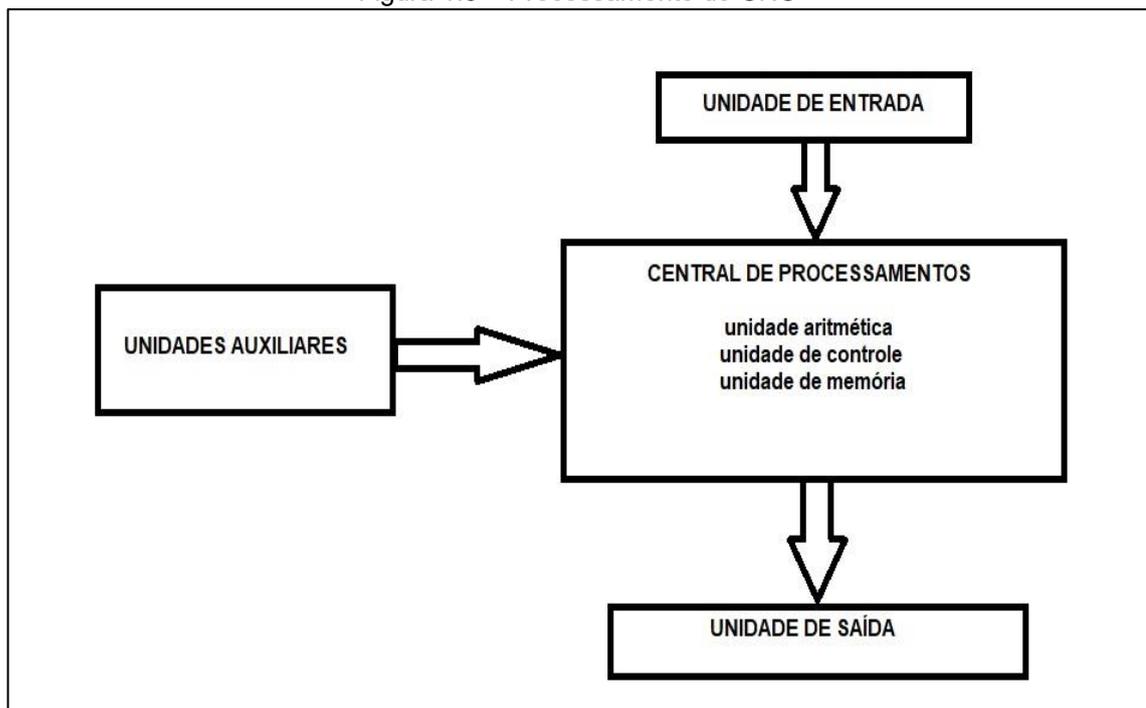
Fonte: www.mundocnc.com.br, 2017

Segundo Rosário (2009), o maior benefício que o CNC proporciona é a automatização do processo e o fato do CNC usar sem a interferência do operador, minimizando os erros de operação. Uma vez definida a tarefa, todas as peças têm o mesmo padrão. Além disso, promove um dinamismo nas peças usinadas, pois a troca de algoritmo é de fácil execução.

Para a criação de um programa em CNC é necessário inserir coordenadas geométricas e parâmetros de usinagem como rotação, velocidade de corte, velocidade de avanço, dados da ferramenta de corte e do material a ser usinado, além dos limites espaciais de trabalho.

O comando numérico computadorizado é uma unidade eletrônica, sendo o computador o elemento responsável por receber a informação numérica, processar essa informação e devolver ao sistema os comandos em forma de sinais para que ocorra a usinagem. Contudo, para que essa operação ocorra, tem que existir entre o computador e a unidade de usinagem um controlador que faz a interface entre as entradas e saídas para a efetivação dos comandos, conforme mostra a Figura 1.5.

Figura 1.5 – Processamento do CNC



Fonte: Autoria própria, 2017

Silva (2015) enfatiza que a unidade de entrada recebe os comandos, identifica-os e os envia para a unidade de processamento. Esta armazena os dados até que estes sejam solicitados pela unidade de saída.

A programação do CNC é uma linguagem que permite ao programador coordenar os movimentos dos eixos para executar uma tarefa. Para executá-las o programador utiliza blocos de comandos. Eles são compostos de letras e números onde as letras indicam a função executada pelo programa e os números o valor que a função representa. A escolha das letras de comando segue um padrão lógico, conforme mostra a Tabela 1.1.

Tabela 1.1 – Bloco de comando com letras

| | |
|----------|--|
| O | Número do programa |
| N | Número de sucessão |
| G | Função preparatória |
| X | Eixo X |
| Y | Eixo Y |
| Z | Eixo Z |
| R | Raio |
| F | Taxa de avanço |
| S | Rotação do fuso |
| H | Compensação de comprimento da ferramenta |
| D | Compensação de raio da ferramenta |
| T | Ferramenta |
| M | Função miscelânea |

Fonte: Autoria própria, 2017

1.3 Tecnologia Laser

O *site* da revista Ciência Hoje (2005) destaca que o princípio básico para a existência do Laser, são as leis fundamentais da interação da radiação luminosa com a matéria. Nesse sentido o Laser é o produto de um fenômeno denominado emissão estimulada, que é a luz amplificada pela emissão de radiação e é emitido por um equipamento denominado projetor laser.

O projetor a Laser é um dispositivo que cria e amplifica uma luz monocromática, em que a disposição da luz liberada é organizada e bem definida, pois se propaga como um feixe.

O corte a Laser se dá a partir de uma alta temperatura concentrada em um feixe de luz, este simplesmente se move ao longo do material, cortando e gravando em peças planas. Através de um *software* é criado um desenho em duas dimensões (2D), que posteriormente é aplicado em diversos tipos de materiais, ideal para chapas finas de metal, madeira, plástico, vidro e cerâmica além de diversas outras aplicações, conforme mostra a Figura 1.6.

Figura 1.6 – Corte a Laser



Fonte: www.gettingsmart.com, 2017

O *site* Seara da Ciência (2015), informa que o laser é utilizado na indústria em atividades que podem ser classificadas em duas categorias: as de processamento de materiais e outras aplicações. O processamento de materiais inclui corte, perfuração, soldagem e, geralmente, envolve o uso de Lasers de alta potência.

As vantagens do corte a laser são várias, dentre as quais se destacam a não necessidade de ferramentas de contato, ao contrário das fresas tradicionais, em que a ferramenta deve ser afiada e, com seu desgaste, um dia substituída. Com o corte a Laser os materiais difíceis de serem cortados podem ser processados mais facilmente. Além disso o laser tem um corte fino que permite realizar tarefas que exigem um maior grau de precisão com desperdício mínimo de material.

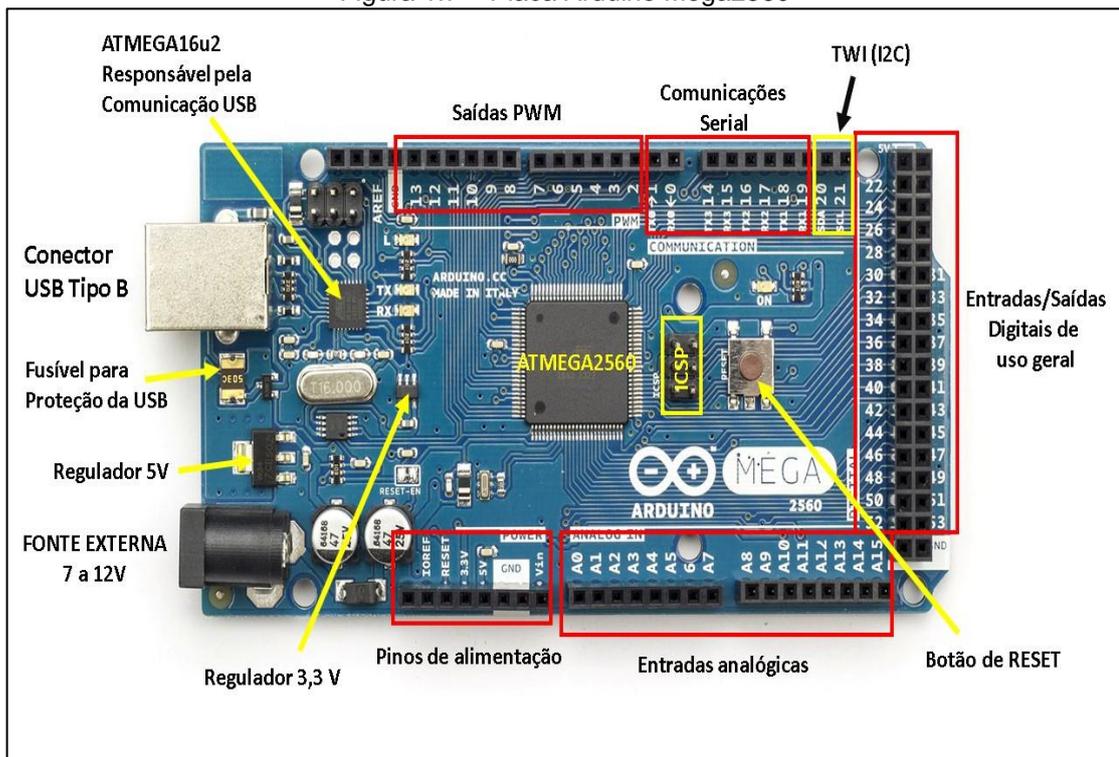
Entretanto, o Laser também possui algumas ressalvas, algumas tarefas como a perfuração de furos de grande diâmetro, são difíceis de realizar. O corte a laser funciona melhor em ambientes sem vibrações, muitas vezes difíceis de conseguir em um ambiente industrial. Os custos iniciais para a implantação de um sistema de corte a Laser podem ser grandes em alguns casos, limitando seu uso em pequenas e médias empresas.

1.4 Arduino Mega2560

Para McRoberts (2011), o Arduino é uma plataforma de placa única onde são inseridas todas as funcionalidades de um controlador, isso permite que sejam desenvolvidos vários projetos, independentes de tipo, modelo ou função. Essa funcionalidade proporciona o desenvolvimento de projetos multidisciplinares.

Ele é constituído de uma placa composta por um microcontrolador Atmel e circuitos de entradas/saídas. Pode ser facilmente conectado a um computador e programado via ambiente de desenvolvimento integrado (IDE) utilizando linguagem baseada em C/C++, sem a necessidade de equipamentos extras além de um cabo USB, a Figura 1.7 mostra uma placa Arduino Mega2560 e seus componentes.

Figura 1.7 – Placa Arduino Mega2560



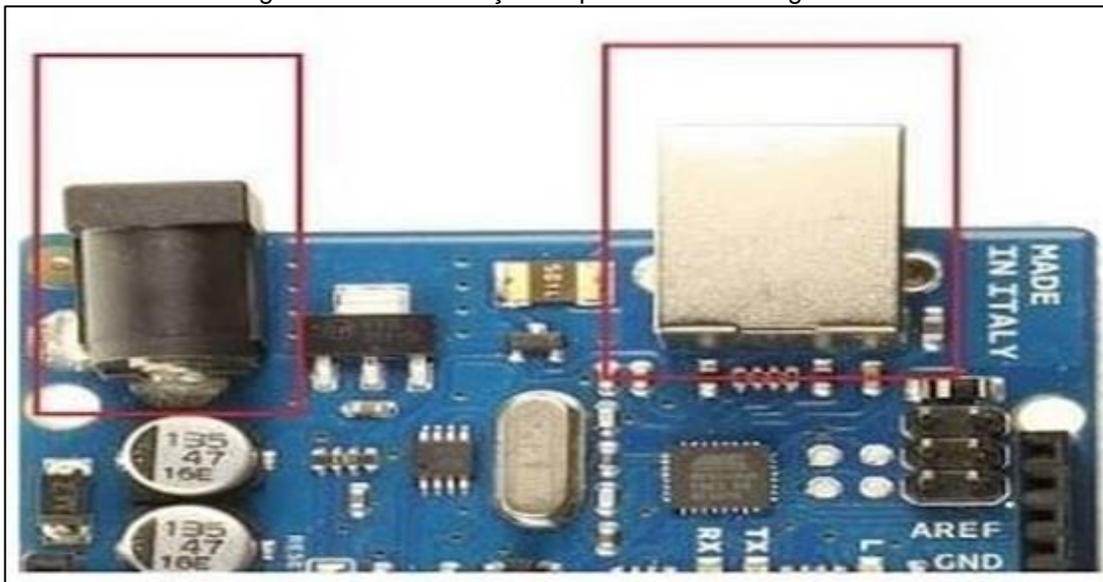
Fonte: www.arduino.cc, 2017

De acordo com Monk (2017), o Arduino Mega2560 foi criado para projetos mais complexos. Esta placa contém um microcontrolador com 54 pinos de entradas e saídas digitais e 15 pinos que podem ser utilizados como modulação de largura de pulso (PWM). Com 16 entradas analógicas, porta serial de *hardware*, oscilador de

cristal de 16 Mega Hertz (MHz), uma conexão universal serial (USB), uma tomada de energia, protocolo de comunicação e um botão de reinicialização.

O Arduino Mega2560 é alimentado através da conexão USB ou com uma fonte de alimentação externa e opera com tensão de 7 a 12 volts (V). É uma placa que contém todas as funcionalidades necessárias para suportar o microcontrolador, conforme mostra a Figura 1.8.

Figura 1.8 – Alimentação da placa Arduino Mega2560



Fonte: www.embarcados.com.br, 2017

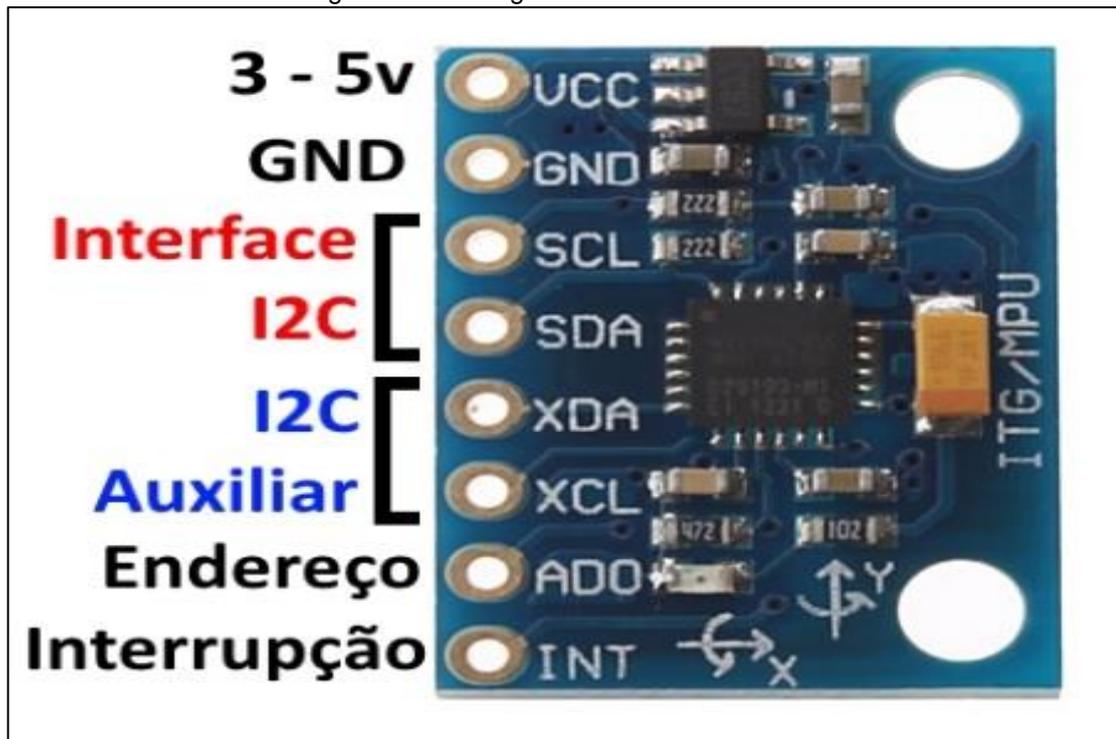
Evans, Noble e Hochenbaum (2013) destacam que além da grande variedade de placas no mercado, existem também as placas chamadas de *Shields* (escudos). Ela tem a função de acrescentar as funcionalidades básicas do Arduino. As placas Shields são fixadas sobre a placa Arduino. As *Shields* também são expansíveis criando módulos com funções diversas.

Os *Shields* mais populares são: motor *Shield*: permite o comando de motores; Protoshield: permite a multiplicação de conexões das portas do Arduino através de uma malha; Ethernet, WiFi, Wireless: permite a comunicação da placa Arduino com a Internet; GPS: permite a verificação de posicionamento geográfico do dispositivo; *Display Shield*: Permite a visualização de informações computadas pelos sensores integrados ao Arduino.

1.5 Sensor MPU-6050

Segundo Ziemann (2018) o CI MPU-6050 possui no mesmo encapsulamento um acelerômetro e um giroscópio de alta precisão. Ele realiza monitoramento em seis eixos, três para o acelerômetro e três para o giroscópio, a Figura 1.9 ilustra a pinagem do sensor MPU-6050.

Figura 1.9 – Pinagem do sensor MPU-6050



Fonte: www.filipeflop.com, 2017

No seu interior o MPU-6050 tem uma função conhecida como DMP (Digital Motion Processor). O DMP processa o algoritmo de detecção de movimento no CI, desta forma o microcontrolador não precisa realizar o processamento. O DMP faz o processamento dos dados do acelerômetro, giroscópio e sensor de temperatura. As informações aparecem no *software* do Arduino, através da aba de monitoramento.

O MPU-6050 contém conversores Analógico/Digital (A/D) de 16 bits de resolução para cada canal, com isso todos os sinais aparecem simultaneamente. Internamente existe uma memória 1024 bytes, onde os valores podem ser armazenados e depois lidos, conforme configuração desejada.

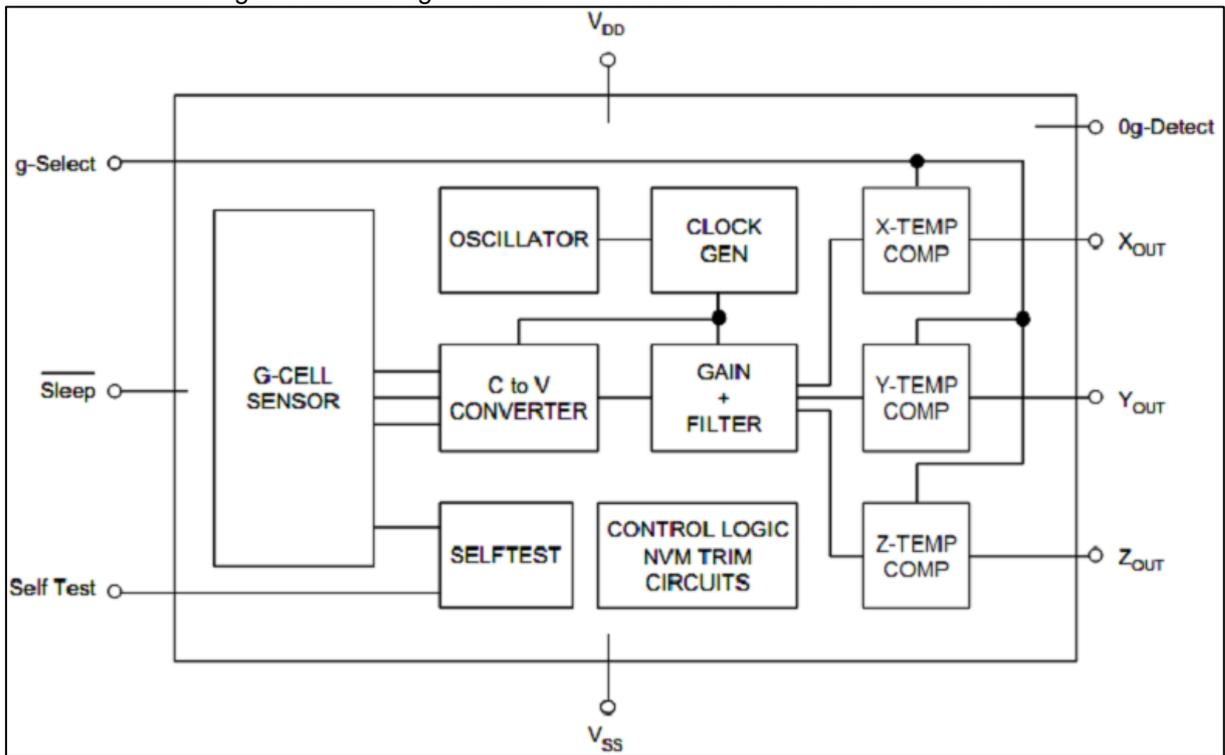
De acordo Zieman (2018) O sensor MPU-6050 possui a função MotionInterface com a capacidade de acompanhar com precisão os movimentos dos usuários com as vantagens de ocupar pouco espaço físico, baixo consumo de energia, alta precisão e repetibilidade, alta tolerância a choques e capacidade de programação de desempenho específica a um custo baixo e a tecnologia MotionTracking pode converter aparelhos em poderosos dispositivos inteligentes 3D.

O MPU-6050 possui três conversores analógico-digitais (ADCs) de 16 bits para a digitalização das saídas do giroscópio e três ADCs de 16 bits para a digitalização das saídas do acelerômetro. Para rastreamento de precisão de movimentos rápidos e lentos, as peças apresentam um giroscópio programável pelo usuário em escala completa de ± 250 , ± 500 , ± 1000 e ± 2000 ° / s (dps) e um acelerômetro programável pelo usuário em escala completa. intervalo de $\pm 2g$, $\pm 4g$, $\pm 8g$ e $\pm 16g$.

Possui um buffer (região de memória) de 1024 Bytes no chip ajuda a reduzir o consumo de energia do sistema, permitindo que o processador do sistema leia os dados do sensor em rajadas e, em seguida, insira um modo de baixo consumo de energia à medida que o MPU coleta mais dados. Com todos os componentes de processamento e sensor necessários no chip com capacidade para suportar muitos casos de uso baseados em movimento, o MPU-6050 possibilita aplicações MotionInterface de baixa potência em aplicações portáteis com requisitos de processamento reduzidos para o processador do sistema.

O MPU-6050 também foi projetado para interagir com múltiplos sensores digitais não inerciais, como sensores de pressão, a comunicação com todos os registros do dispositivo é realizada usando I2C a 400kHz. A peça apresenta uma robusta tolerância a choques de 10.000 g e possui filtros passa-baixo programáveis para os giroscópios, acelerômetros e sensor de temperatura no chip. A Figura 1.10 traz o diagrama de blocos do funcionamento do sensor acelerômetro MPU-6050.

Figura 1.10 – Diagrama de blocos de funcionamento do MPU-6050



Fonte: www.eletrondex.blogspot.com, 2018

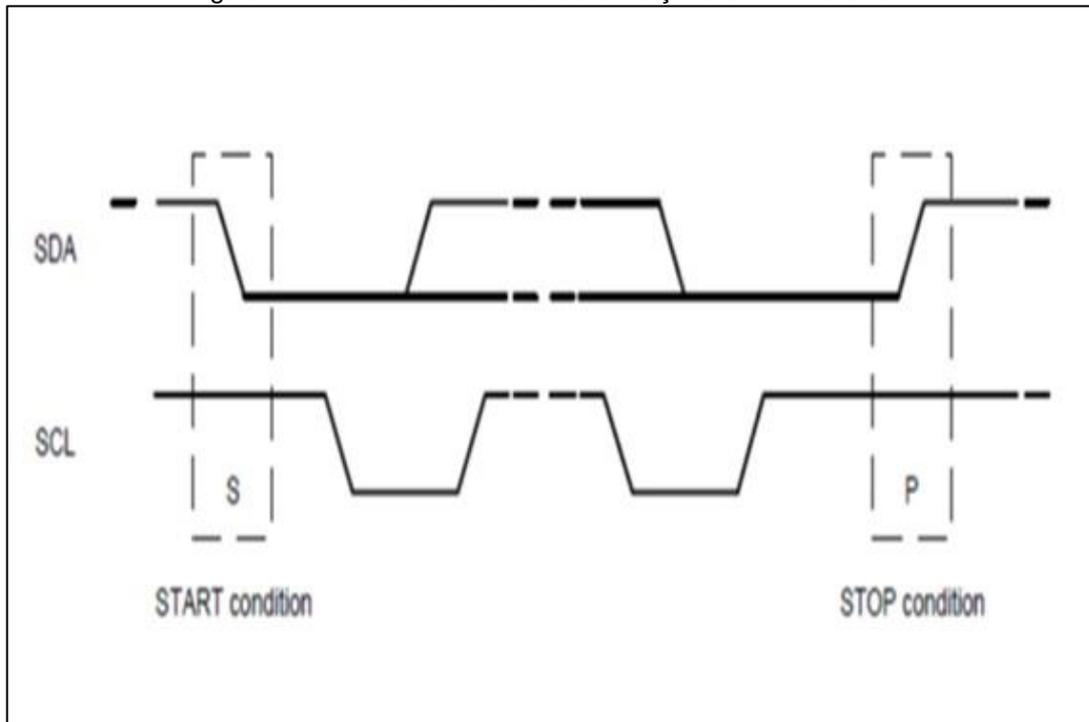
O I2C é uma interface de dois fios composta pelos dados de sinais de série (SDA) e relógio serial (SCL). Em uma implementação de interface I2C de forma generalizada, os dispositivos conectados podem ser um mestre ou um escravo. O dispositivo mestre coloca o endereço do escravo no barramento e o dispositivo escravo com o endereço correspondente reconhece o mestre.

O MPU-6050 sempre opera como um dispositivo escravo quando se comunica com o processador do sistema, que atua como mestre. O endereço do escravo do MPU-6050 é o b110100X, que tem 7 bits de comprimento. O bit LSB do endereço de 7 bits é determinado pelo nível lógico no pino AD0. Isso permite que dois MPU-6050s sejam conectados ao mesmo barramento I2C. Quando usado nesta configuração, o endereço de um dos dispositivos deve ser b1101000 (o pino AD0 é lógico baixo) e o endereço do outro deve ser b1101001 (o pino AD0 é lógico alto).

A comunicação no barramento I2C é iniciada quando o mestre coloca a condição START (S) no barramento, que é definida como uma transição ALTA para BAIXA da linha SDA, enquanto a linha SCL está ALTA. O barramento é considerado ocupado até que o mestre coloque uma condição STOP (P) no barramento, que é

definida como uma transição BAIXA para ALTA na linha SDA enquanto a SCL está ALTA e o barramento permanece ocupado se um START (Sr) repetido for gerado em vez de uma condição STOP. Abaixo a ilustração dessas condições na Figura 1.11.

Figura 1.11 – Funcionamento das condições START e STOP



Fonte: MPU-6000 Datasheet, 2017

1.6 *Design* de interação e sistemas supervisórios

Preece, Rogers e Sharp (2005) enfatizam que o *design* de interação é o desenho de produtos interativos que fornecem suporte às atividades cotidianas das pessoas, seja no lar ou no trabalho. Especificamente significa criar experiência que melhorem e entendam a maneira como as pessoas trabalham, comunicam e interagem.

A partir da necessidade e importância de se entender como os usuários agem e reagem a situações e como se comunicam e interagem, acarreta o envolvimento de diversos profissionais de diferentes disciplinas, como psicólogos e sociólogos em questões referentes ao *design* de interação. Ele consiste em encontrar maneiras de

fornecer suporte às pessoas e contrasta com a engenharia de *software*, que enfoca principalmente a produção de soluções para determinadas aplicações.

Preece, Rogers e Sharp (2005) esclarecem que todas essas questões devem ser analisadas antes de serem executadas na criação de uma interface com o usuário. Para que as metas de usabilidade possam ir além da eficiência, uma boa interface tem que ser também, intuitiva, confiável e segura, além disso, uma interface eficiente deve ser agradável, interessante e esteticamente apreciável, conforme ilustra a Figura 1.12.

Figura 1.12 – Interfaces do *design* de interação

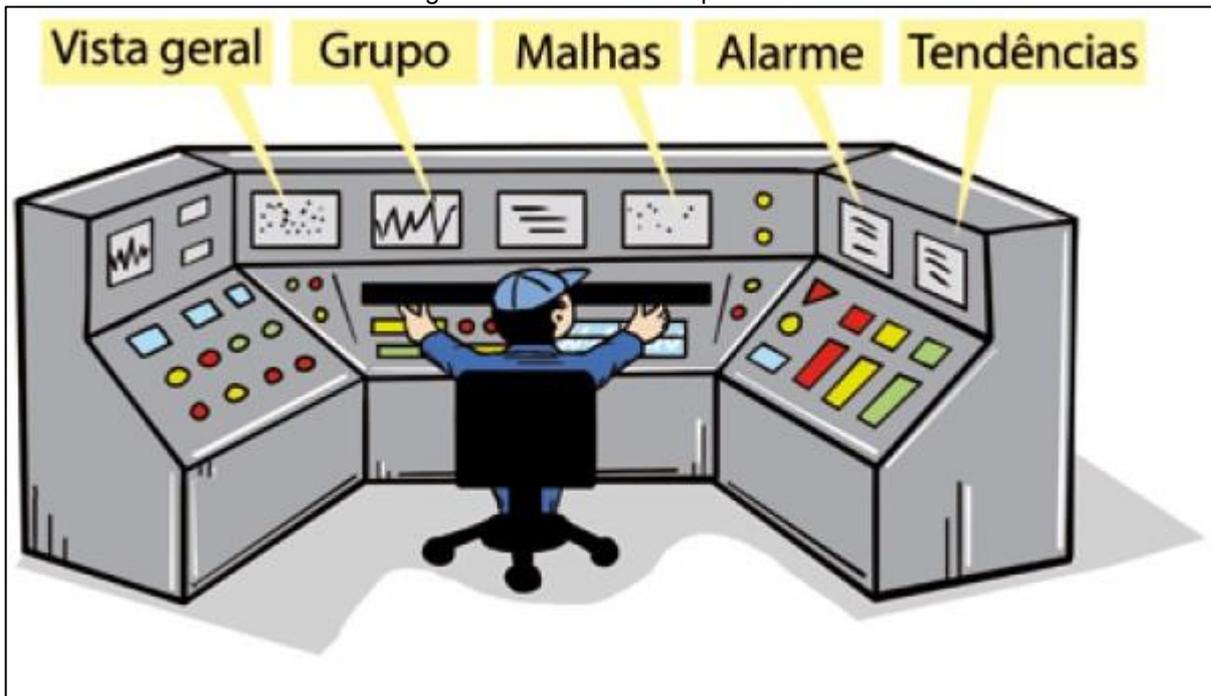


Fonte: www.imasters.com.br, 2017.

Para o design de interação é importante reconhecer e entender o equilíbrio entre metas de usabilidade e os resultados decorrentes da experiência do usuário, as diferentes combinações dessas metas, desde que sejam compatíveis com o objetivo, devem ser consideradas para atender as necessidades dos usuários.

Na opinião de Rosário (2009) um sistema supervisor é uma ferramenta destinada a captar e reter em um banco de dados, as informações de um processo. Os dados são enviados por sensores que retêm as informações denominadas variáveis de processo de acordo ilustra a Figura 1.13.

Figura 1.13 – Sistema supervisório



Fonte: www.mecatronizando.com.br, 2018.

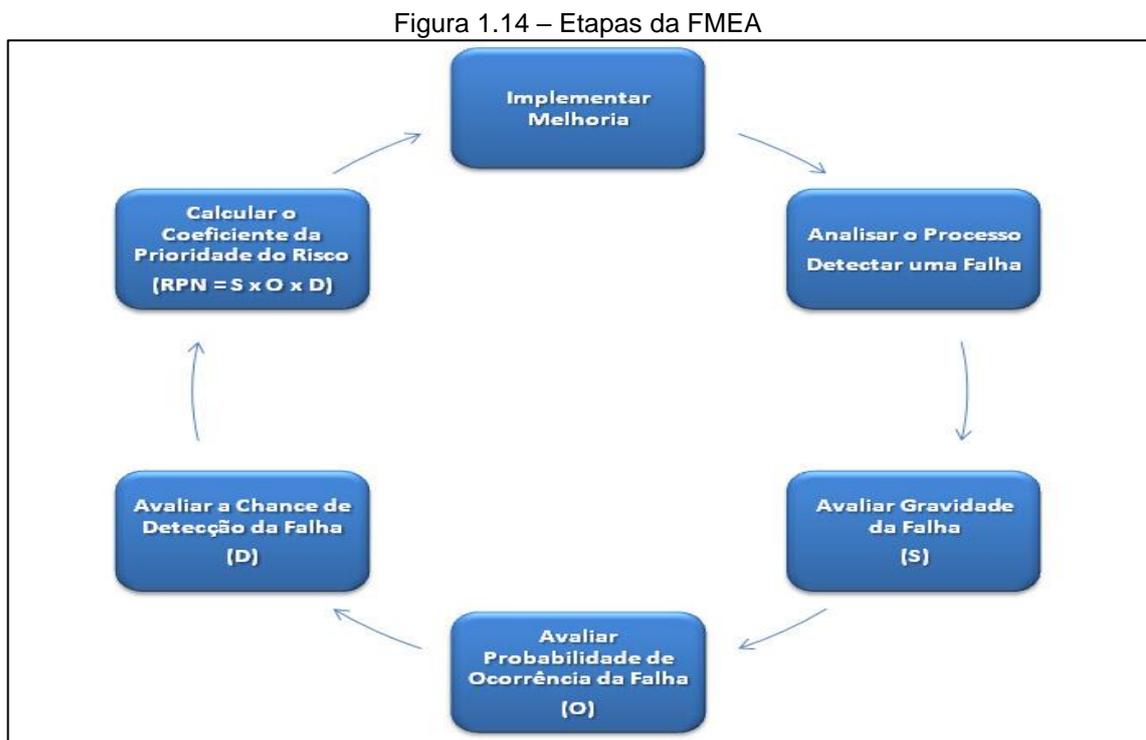
Rosário (2009) informa que os sistemas supervisórios são especificados como IHM ou controle supervisório e aquisição de dados (SCADA). E dentre eles os mais utilizados no mercado são: Elipse®; FactoryTalk View SE®; iFIX®; InduSoft Web Studio®; ProcessView®; ScadaBR®; SIMATIC Wincc®; Vijeo Citect®; Wondeware inTouch®.

As principais vantagens na utilização de um sistema supervisório no processo são: Análise de tendências que buscam no banco de dados informações que criam previsões para ações futuras. Alarmes que sinaliza em tempo real qualquer falha no equipamento. Operação remota a partir da sala de controle. Geração de relatórios para análise e tomada de ação.

1.7 Plano de manutenção – FMEA

A Análise do Modo e Efeito de Falha, conhecida como FMEA (do inglês *Failure Mode and Effect Analysis*), é uma ferramenta usada no desenvolvimento de um plano de manutenção preventivo.

Fogliato e Ribeiro (2009) destacam que o objetivo da FMEA é identificar potenciais modos de falha de um produto ou processo de forma a avaliar o risco associado a estes. São classificados em termos de importância, recebem ações corretivas com o intuito de diminuir as incidências de falhas, além disso, sua aplicação impacta diretamente no retorno financeiro da empresa que é decorrente da minimização e eliminação de falhas potenciais nos processos produtivos, conforme mostra a Figura 1.14.



Fonte: www.scielo.br, 2017

A ferramenta FMEA é um documento de apoio que dá direcionamento à manutenção que requer constantes revisões de atualização. Isso é inerente de uma metodologia constante, no aperfeiçoamento da qualidade dos produtos e serviços de uma organização. Ainda, assegura que os métodos de inspeção e análise sejam feitos de forma padrão com a criação de históricos de defeitos existentes, bem como os futuros, auxilia no planejamento da produção, controle da qualidade e a engenharia de processo com planos de controle.

A FMEA começou a ser utilizada na década de 1940, pelas forças armadas norte-americanas. Em 1970 começou a ser aplicada na indústria automobilística para melhorias em questões de análise de regulamentação, segurança, *design* e produção.

Para Dalosto, Balestrassi e Peruchi (2015) a FMEA analisa primeiro os componentes mais simples, as causas do defeito e como ele interfere em outros componentes mais complexos. Chega-se a essa conclusão questionando de que maneira um componente pode falhar, os tipos de falhas que são observadas, partes do sistema afetado e a importância de como preveni-las.

Palady (2004) aponta que para cada tipo de falha, a equipe responsável calcula o coeficiente da prioridade do risco (RPN), que varia de 1 a 10. Quanto maior o produto pior é a falha descrita. As perguntas que devem ser feitas para se chegar aos valores são:

- ocorrência: qual a chance desse tipo de falha ocorrer?
- gravidade: se a falha ocorrer, qual a gravidade do dano que pode ser gerado?
- detecção: se a falha ocorrer, qual a chance de ser detectada?

Se a ocorrência for alta deve-se eliminar as causas, adicionando verificações como dupla-chechagem. Se a detecção for baixa, deve-se verificar eventos que ocorreram antes da falha para que a nova etapa considere a possibilidade de automatizar o processo.

A filosofia de prevenção de falhas por meio de um plano de manutenção aumenta a confiabilidade, qualidade, segurança, diminuição do custo, tempo de desenvolvimento, planejamento, superação das expectativas dos clientes, padronização do conhecimento tecnológico, sinergia entre engenharia do produto, manufatura e produção, identificando e solucionando os problemas que acontecem no acompanhamento dos riscos.

2 METODOLOGIA

Neste capítulo encontra-se a trajetória para o desenvolvimento e construção do projeto que se intitula Automação do plano de manutenção da máquina CNC Laser. Trata-se de uma pesquisa aplicada que é desenvolvida nas dependências da FATEC São Bernardo do Campo.

Prodanov e Freitas (2013) destacam que a metodologia é o caminho a percorrer para o desenvolvimento de uma pesquisa. Enfocam que os métodos são procedimentos amplos do raciocínio e as técnicas são procedimentos que operacionalizam os métodos mediante instrumentos adequados.

Severino (2017) enfatiza que a preparação metódica e planejada de um trabalho científico supõe uma sequência de etapas que compreende: determinação do tema-problema e justificativa; levantamento da bibliografia referente ao tema; leitura e documentação dessa bibliografia após seleção; construção lógica do trabalho é redação do texto.

A preparação metódica está diretamente ligada ao tipo de pesquisa que se pretende realizar, utilizando técnicas que estão em sintonia com o assunto abordado. As técnicas dão base para a forma correta da elaboração do projeto, que são procedimentos que encadeiam o raciocínio para uma metodologia de aplicação prática.

A construção da redação do TCC tem como base o Manual de Normalização de Projeto de Trabalho de Graduação da Fatec-SBC (2017), que se encontra embasado nas normas da ABNT. É escrito em uma linguagem simples, concisa é com terminologia adequada.

2.1 Tema-problema com justificativa e descrição funcional do projeto

O objetivo desse trabalho que se intitula Automação do plano de manutenção da máquina CNC laser é montar uma máquina CNC com corte a laser e implantar os dados do plano de manutenção preventivo criado à partir do uso da ferramenta FMEA, para automatizar o acesso as informações referentes às ações recomendadas de um plano preventivo da máquina. A confirmação da viabilidade do projeto vai possibilitar o mesmo tipo de automação em máquinas diversas nos segmentos industriais, reduzindo os custos operacionais de uma planta e facilitar a sua gestão.

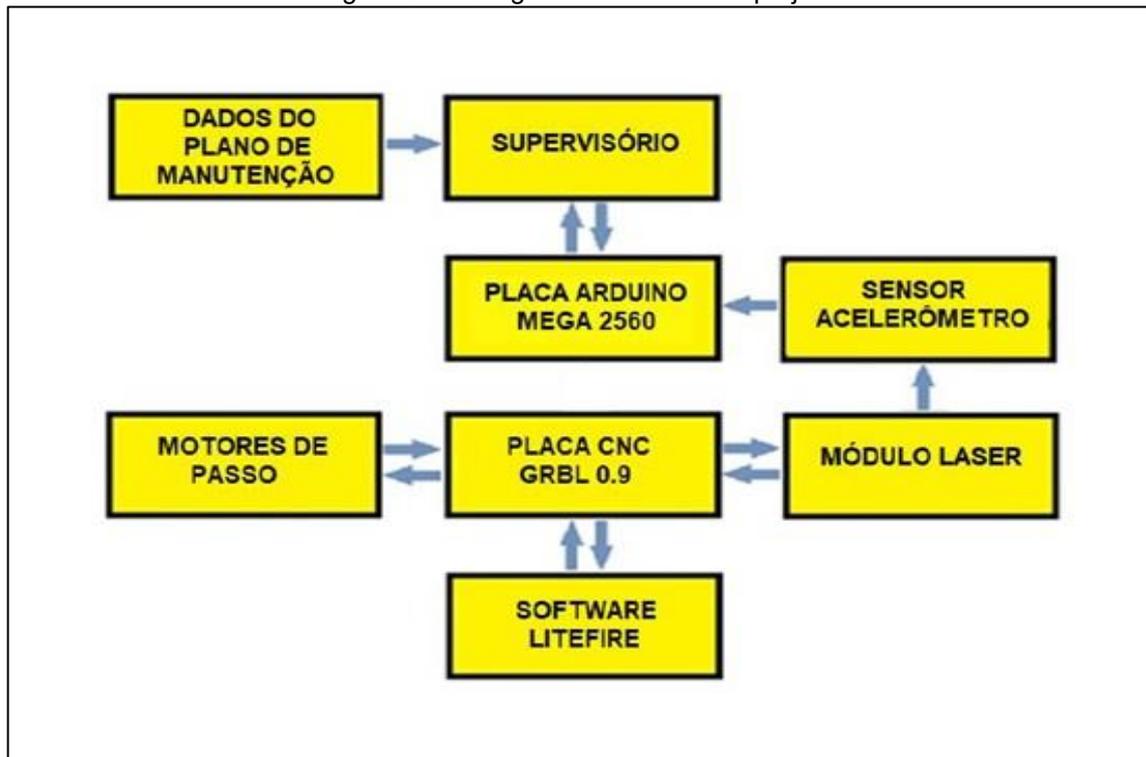
Justifica-se pela redução de gasto com mão de obra, aumento da disponibilidade da máquina, dispensa a análise de vibração frequente por meio do colaborador com uso de aparelhos de medição, reduz o custo, o estoque de peças e armazenamento de materiais no almoxarifado, uma vez que é conhecido em tempo real o momento do início do ponto de falha.

O investimento não é alto, se comparado às vantagens oferecidas no acompanhamento do funcionamento da máquina, e as informações de alertas ou avisos do plano de manutenção informados, conforme o plano padroniza os métodos na execução das atividades.

O sistema é utilizado nas indústrias automobilísticas, construção de máquinas, alimentícia, têxtil, farmacêutica e cosméticos e é uma ideia convergente com os fundamentos da indústria 4.0. Esses são os motivos que tornam o projeto viável, promissor e com grande possibilidade de aceitação pelo mercado.

Para melhor visualização sobre o projeto é apresentado o diagrama de blocos, conforme mostra a Figura 2.1.

Figura 2.1 – Diagrama de blocos do projeto



Fonte: Autoria própria, 2017

O sistema é composto por um módulo sensor que indica a aceleração durante o deslocamento no local de monitoração e controle, transmitindo dados para uma placa de controle onde as informações são processadas conforme programação e transmitidos através de um *software* que faz a função de uma interface homem-máquina informando em tempo real a situação local do componente e ou conjunto.

Para ocorrer a automação do plano de manutenção, as informações extraídas do plano de manutenção que é desenvolvido através da metodologia FMEA, com base em informações retiradas de manuais e da experiência dos manutentores para realizar uma determinada ação de prevenção, as quais são usadas na criação do sistema supervisório. Ele é alimentado com base nas ações recomendadas do plano de manutenção e aparecem em forma de mensagens conforme o sensor acelerômetro atingir o limite do valor pré-configurado.

As informações captadas pelo sensor por meio do cabo de comunicação ligado a placa de controle Arduino Mega2560 são enviadas para *software* do sistema supervisório, possibilitando a sua visualização de funcionamento e a mensagem do plano de manutenção com a tomada de ação de correção pelo operador do sistema.

2.2 Etapas teóricas e práticas para o desenvolvimento do projeto

Após delimitar o tema-problema com justificativa e descrição do projeto parte-se para as seguintes etapas:

Primeira etapa: reuniões semanais com o orientador, para esclarecimento das dúvidas com relação a fundamentação teórica e acompanhamento do desenvolvimento lógico do projeto.

Segunda etapa: levantamento bibliográfico, pesquisas em *sites* especializados, livros de autores nacionais e internacionais, manuais e catálogos de empresas e consultas a empresas fabricantes de componentes.

Terceira etapa: seleção dos conteúdos para construir o capítulo 1 - Fundamentação teórica.

Quarta etapa: estudo da viabilidade econômica dos materiais para construção do projeto. Pesquisas de preços em lojas especializadas e em *sites*. Aquisição conforme Tabela 2.1.

Tabela 2.1 – Materiais para confecção do projeto

| Número da peça | Nome do componente | Quantidade | Custo |
|-------------------------------|---|------------|---------------|
| 1 | FUSO AÇO INOX 304 M8 X 1000 mm (PASSO 2mm) | 1 | 20,00 |
| 3 | ROLAMENTO LINEAR ESFERAS LM10UU | 8 | 48,00 |
| 4 | ROLAMENTO LINEAR ESFERAS LM8UU | 4 | 20,00 |
| 5 | CASTANHA TR8 diâmetro 8 X 2 mm | 3 | 7,50 |
| 6 | ACOPAMENTO DO FUSO / MOTOR diâmetro 8 mm | 3 | 4,50 |
| 7 | T-PORCA 20-M5 P/ PERFIL ALUMINIO 2020 | 76 | 12,16 |
| 8 | BLOCO PORCA QUADRADA T 2020 M5 X 14 mm | 14 | 7,00 |
| 9 | HASTE DE AÇO INOX diâmetro 10 X 1000 mm | 1 | 18,00 |
| 10 | HASTE DE AÇO INOX diâmetro 8 X 1000 mm | 1 | 15,00 |
| 11 | SUPORTE 90° ALUMINIO 3030 28 X 35 | 8 | 24,00 |
| 12 | PERFIL DE ALUMINIO 20 X 20 X 2000 mm | 1 | 25,00 |
| 13 | MOLA HELICOIDAL diâmetro 10 mm; # 8mm passo: 4 mm Compr. 30 mm | 2 | 8,00 |
| 14 | DISSIPADOR DE CALOR quadr. 9 X 5 mm | 3 | 3,00 |
| 15 | BASE DE ALUMINIO 180 X 250 15 mm | 1 | 50,00 |
| 16 | MOTOR DE PASSO JLB 17HS1352-P4130 compr. 34 mm ; 1,33 A; 12V; 0,25 N/m | 3 | 75,00 |
| 17 | PLACA DE CONTROLE GRBL 0,9J, PORTA USB | 1 | 60,00 |
| 18 | FONTE ADAPTADOR PA-24135-089 input: 100-240v; 2,5a (50-60HZ); output: 24V | 1 | 32,00 |
| 19 | GRAMPO DE FIXAÇÃO BASE 12 X 20 X 50 mm | 4 | 120,00 |
| 20 | BLOCO TRAVESSEIRO LINEAR EIXO DO SUPORTE diâmetro 8mm | 8 | 56,00 |
| 21 | ROLAMENTO 688 ZZ | 2 | 4,00 |
| 22 | PLACA DE EXPANSÃO MEGA 2560 R3 | 1 | 38,00 |
| 23 | PLACA DE PROTEÇÃO CONTRA CURTO RAMPS 1.4 | 1 | 22,00 |
| 24 | MODULO SENSOR DE VIBRAÇÃO TIPO NORMALMENTE ABERTO SW18010P | 3 | 33,00 |
| 25 | MAQUINA DE GRAVURA DO LASER AZUL 3000 Mw, 450NM | 1 | 170,00 |
| 26 | OCULOS DE PROTEÇÃO DE LASER | 1 | 10,00 |
| 27 | COOLER DO MODULO LASER | 1 | 12,00 |
| CUSTO TOTAL DO PROJETO | | | 894,16 |

Fonte: Autoria própria, 2017

Quinta etapa: é elaborado um planejamento para construção do protótipo com início em janeiro de 2018. O planejamento para a construção do projeto é apresentado no cronograma da Tabela 2.2.

Tabela 2.2. – Cronograma do desenvolvimento do projeto

| Atividades | Semanas | | | | | | | | | | | | | | | | | | | |
|-------------------------------|---------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Compra dos materiais | █ | | | | | | | | | | | | | | | | | | | |
| Montagem do CNC | | █ | █ | | | | | | | | | | | | | | | | | |
| Criação do supervisor | | | █ | █ | █ | █ | █ | █ | █ | █ | █ | | | | | | | | | |
| Testes do protótipo | | | | | █ | █ | █ | █ | █ | █ | █ | █ | █ | █ | | | | | | |
| Exposição do protótipo | | | | | | | | | | | | | | | █ | | | | | |
| Criar apresentação para banca | | | | | | | | | | | | | | | | █ | █ | | | |
| Apresentação banca | | | | | | | | | | | | | | | | | | | █ | |
| Efetuar correções da banca | | | | | | | | | | | | | | | | | | | | █ |

Fonte: Autoria própria, 2018

Sexta etapa: montagem mecânica que se refere as estruturas de alumínio, ajustes para que elas possam ter as mesmas dimensões, montagens dos eixos, cabeçote, mesa de trabalho, motores de passo.

Sétima etapa: instalação e configuração da placa de controle GRBL do CNC, que se refere a parametrização dos motores de passo e testes da placa.

Oitava etapa: testes preliminares de funcionamento da máquina CNC no que tange a verificação dos eixos manualmente, para detectar possíveis erros no funcionamento mecânico fazendo o uso do *software* LiteFire.

Nona etapa: desenvolvimento do supervisor na criação de telas. Comunicação da tela com a placa de controle Arduino Mega2560 e o sensor acelerômetro.

Décima etapa: configuração da placa Arduino Mega2560. Desenvolvimento de um programa em linguagem C e compilação dos dados com a placa de controle

Arduino Mega2560. Ligação do acelerômetro a placa Arduino Mega2560 e verificação do sensor. Comunicação do supervisor com a placa Arduino Mega2560.

Décima primeira etapa: desenvolvimento e integração do plano de manutenção, criação de um plano de manutenção para padronizar as ações recomendadas.

Décima segunda etapa: testes de funcionamento do projeto no que tange a verificação do funcionamento do motor com o uso do *software* LiteFire. Substituição do motor pelo módulo laser e testes.

Décima terceira etapa: relação dos erros encontrados no desenvolvimento do projeto e possíveis soluções.

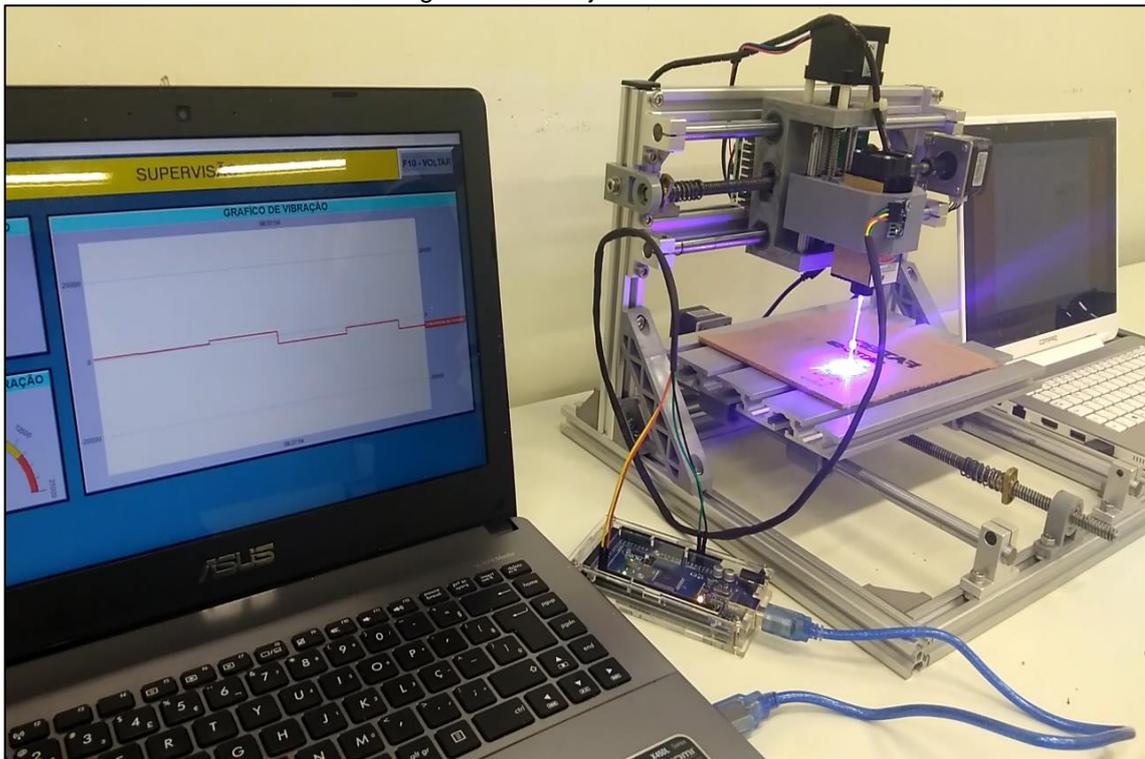
Décima quarta etapa: após o desenvolvimento do projeto faz-se as Considerações Finais e Resumo.

3 DESENVOLVIMENTO DO PROJETO

Neste capítulo encontra-se o desenvolvimento e construção do projeto passo a passo intitulado Automação do plano de manutenção da máquina CNC Laser.

Buscando melhor visualização e compreensão do projeto, a Figura 3.1 ilustra o projeto finalizado.

Figura 3.1 – Projeto finalizado



Fonte: Autoria própria, 2018

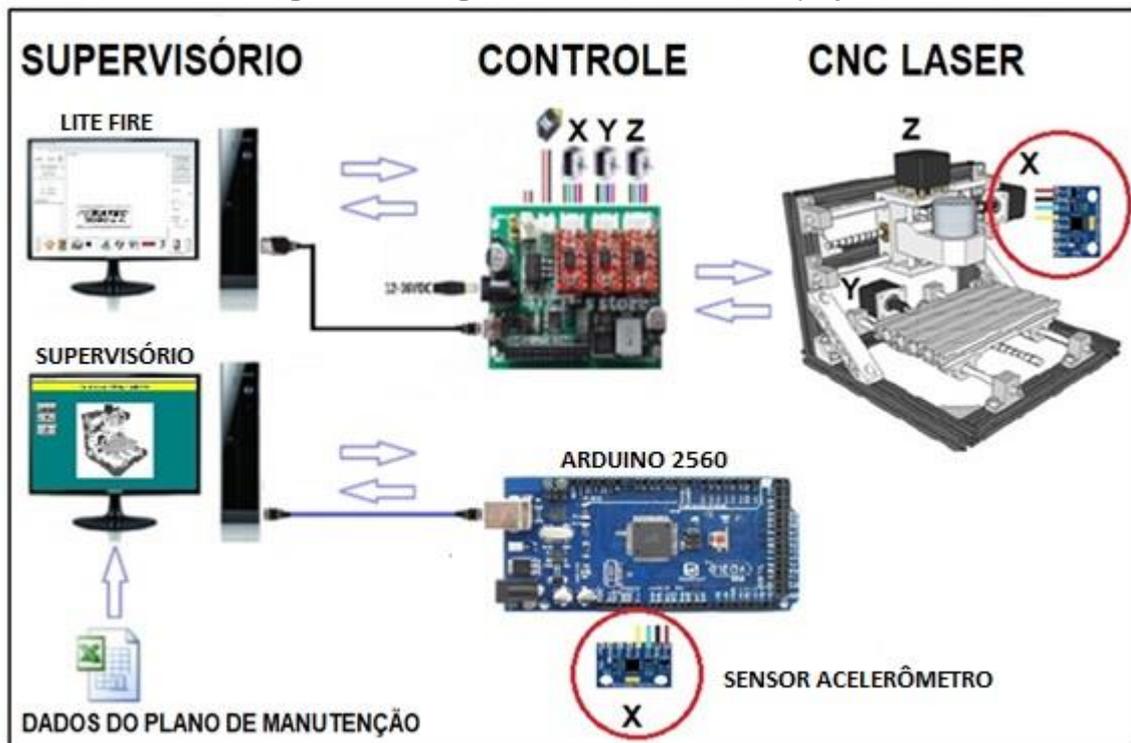
Para construção do dispositivo são utilizados perfisados de alumínio para montagem da estrutura da máquina CNC, fusos roscados para a movimentação da máquina em três eixos (X, Y e Z), motores de passo, dispositivo de gravura a laser (projektor laser), placa GRBL de controle do CNC, placa de controle Arduino Mega2560 e sensor acelerômetro MPU-6050.

O funcionamento do projeto se dá da seguinte maneira, insere-se os dados do plano de manutenção no supervisor, em seguida se estabelece os limites de vibrações. Quando os limites são atingidos geram alarmes, que por sua vez indicam

ações recomendadas para eliminar as falhas da máquina CNC. A máquina CNC é controlada por dois sistemas, o supervisor que verifica falhas de vibração e o de controle dos eixos que comanda os movimentos.

O ciclo de trabalho completo do protótipo inicia-se com a alimentação da máquina CNC, abertura do *software* de supervisão do Eclipse SCADA® e a abertura do *software* LiteFire controlador dos motores de passo e projetor laser. Através do LiteFire uma figura é criada. O laser é ligado pelo comando *Start* do programa e inicia-se a sua impressão. Se durante a gravura de figura a variação de vibração atingir os valores máximos permitidos, um aviso aparece na tela do supervisor juntamente com a ação recomendada para a falha, o diagrama do funcionamento do projeto é mostrado na Figura 3.2.

Figura 3.2 – Diagrama de funcionamento do projeto



Fonte: Autoria própria, 2018

O desenvolvimento do projeto envolve a seguintes etapas:

- Montagem mecânica da máquina, ajustes e alinhamento;
- Instalação e configuração da placa de controle do CNC;
- Teste preliminar de funcionamento da máquina CNC;

- Desenvolvimento do supervisor com o *software* Elipse SCADA;
- Configuração da placa Arduino controladora do supervisor;
- Comunicação do sensor de vibração com a com o supervisor;
- Desenvolvimento e Integração do plano de manutenção;
- Testes finais e funcionamento do projeto;
- Obstáculos e soluções durante o desenvolvimento do projeto.

3.1 Montagem mecânica da máquina, ajustes e alinhamento

Para o bom funcionamento a máquina CNC requer uma estrutura leve e resistente para absorver as vibrações, movimentações em todos os eixos com estrutura para suportar todas os componentes que a compõem.

Dando início, seleciona-se uma barra de perfilado de alumínio com dimensões de 20 x 20 x 2000 milímetros (mm) que tem a finalidade de confeccionar a estrutura da máquina CNC. Os perfilados são cortados nas medidas de 2 x 220 mm por 2 x 240 mm para confecção da base. O encosto vertical que suporta a placa controladora, o projetor laser e os motores dos movimentos X e Z, tem as medias de 2 x 220 mm por 2 x 200 mm, a estrutura do CNC montada é ilustrada na Figura 3.3.

Figura 3.3 – Estrutura do CNC montada

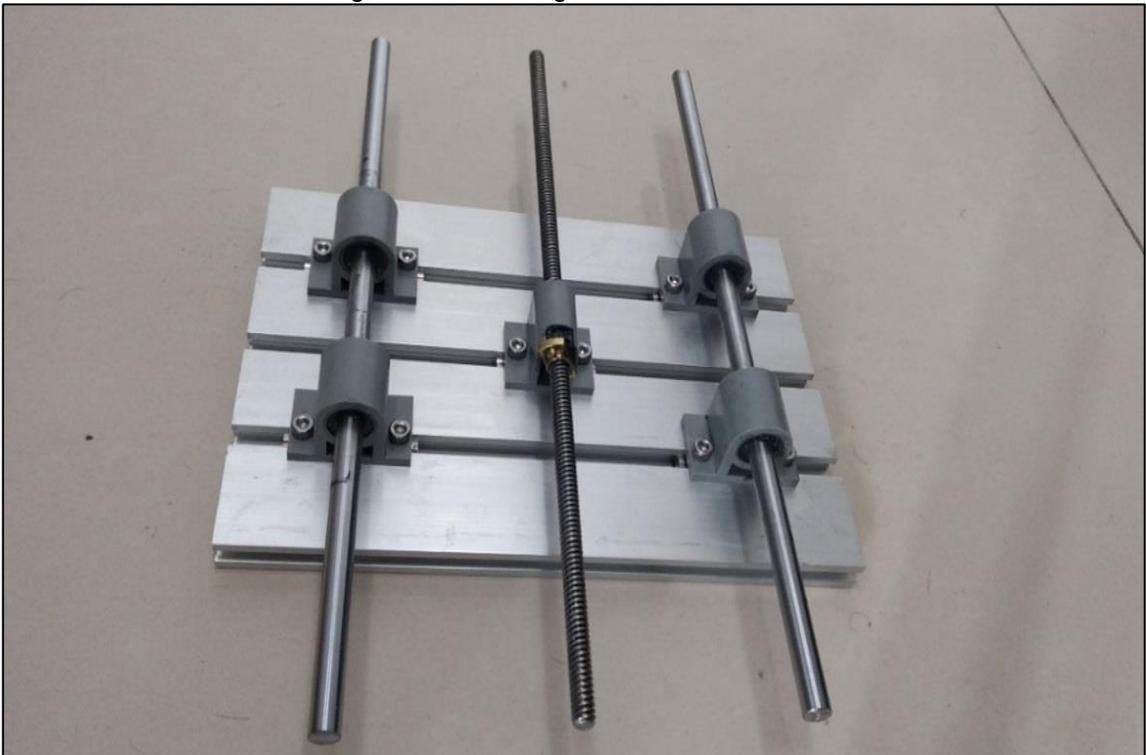


Fonte: Autoria própria, 2018

Após a confecção e montagem dos suportes de alumínio utiliza-se seis eixos de aço liso com 8 mm de diâmetro, que tem a função de guia linear. Os motores são acoplados a fusos de aço inox com 8 mm de diâmetro e 2 mm de passo.

A mesa do CNC é montada unindo-se barras chatas de perfilado de alumínio, as medidas da mesa são 100 x 160 mm. Na base inferior da mesa são instalados os mancais e das barras de guia e do fuso roscado como mostra a Figura 3.4.

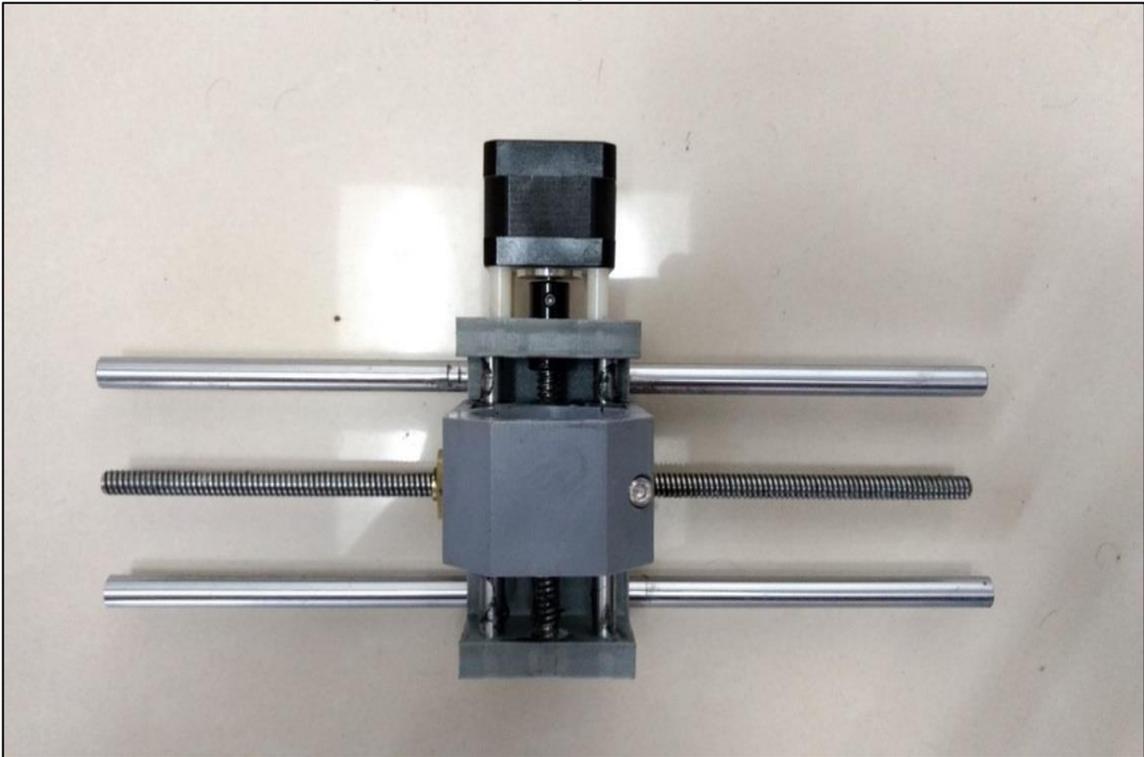
Figura 3.4 – Montagem da mesa do CNC



Fonte: Autoria própria, 2018

O Carro móvel, onde fica instalado o projetor de corte laser, tem um suporte que permite o carro movimentar nos eixos X e Z. O suporte possui duas guias lineares e duas barras roscadas para realização dos seus movimentos, o conjunto montado é ilustrado na Figura 3.5.

Figura 3.5 – Montagem do carro móvel do CNC



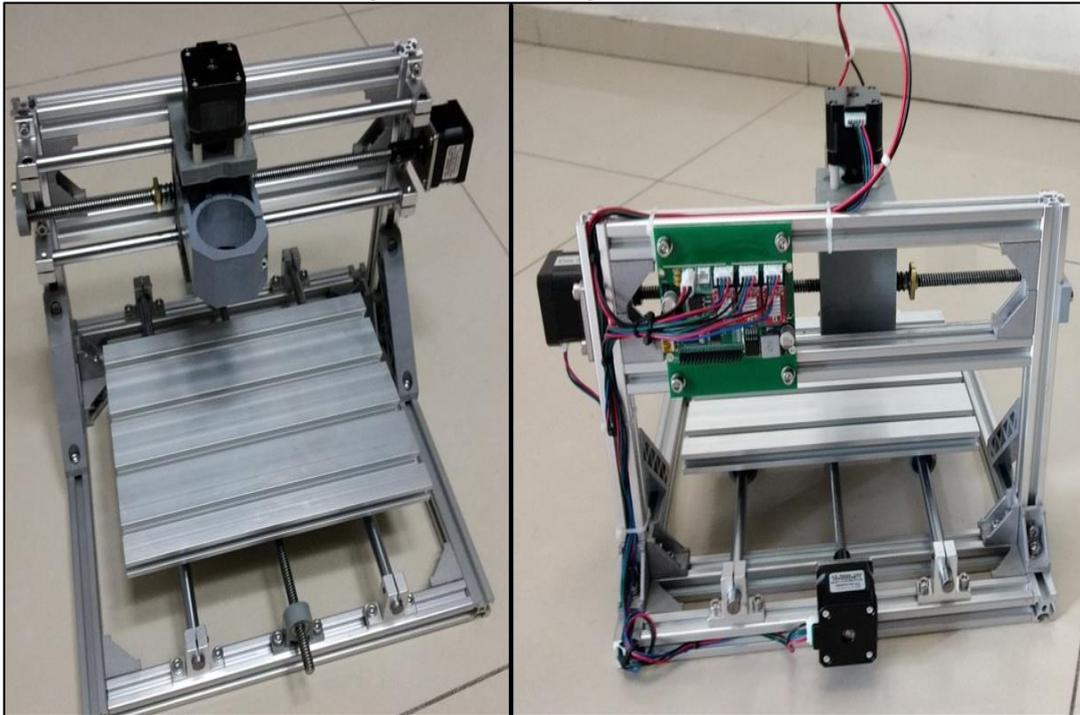
Fonte: Autoria própria, 2018

Para a fixação dos parafusos ao conjunto utiliza-se trava química para evitar folgas ou desmontagem das peças com a vibração do conjunto.

O movimento da máquina nos três eixos é realizado através de três motores de passo com alimentação de 12 Volts (V) e corrente de 1,33 Amperes (A), gerando um torque nominal de 0,25 Newton metros (Nm).

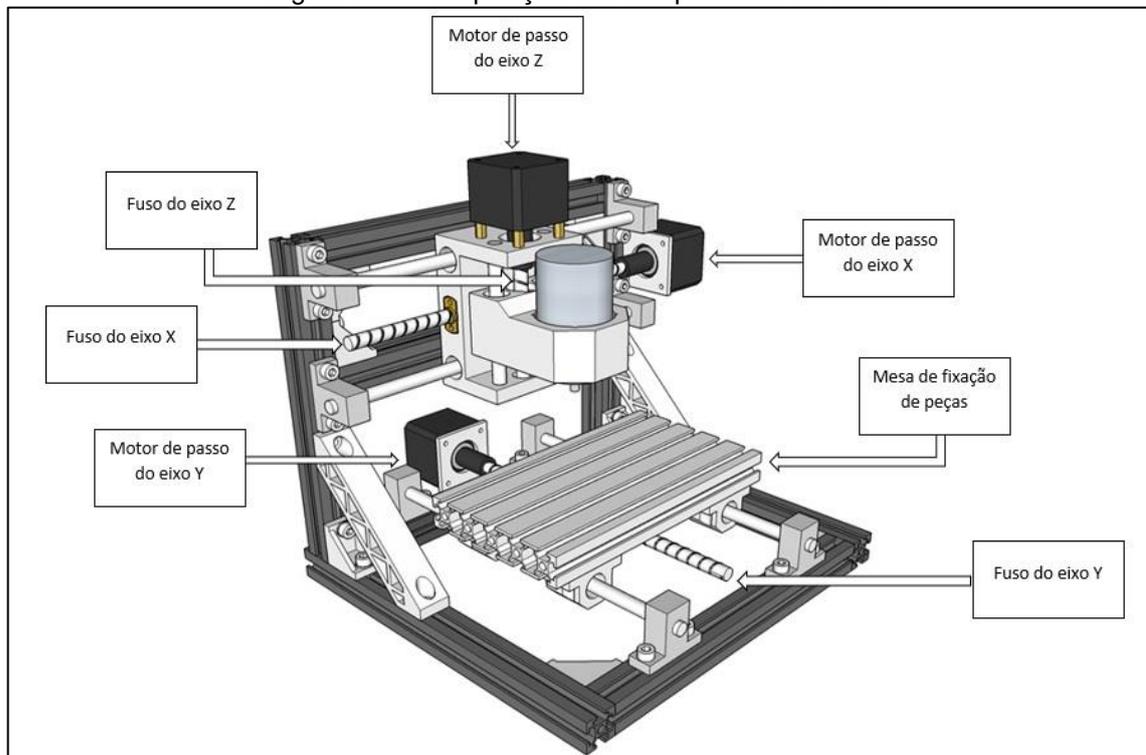
Com todas as peças que compõem o CNC montadas é feita a união das partes chegando a fase final de montagem do equipamento. O protótipo após montado fica com dimensões compactas e de fácil transporte. A Figura 3.6a ilustra a finalização da montagem do CNC com vistas frontal e posterior e a Figura e 3.6b a disposição de cada componente da máquina.

Figura 3.6a – Montagem finalizada do CNC



Fonte: Autoria própria, 2018

Figura 3.6b – Disposição dos componentes do CNC



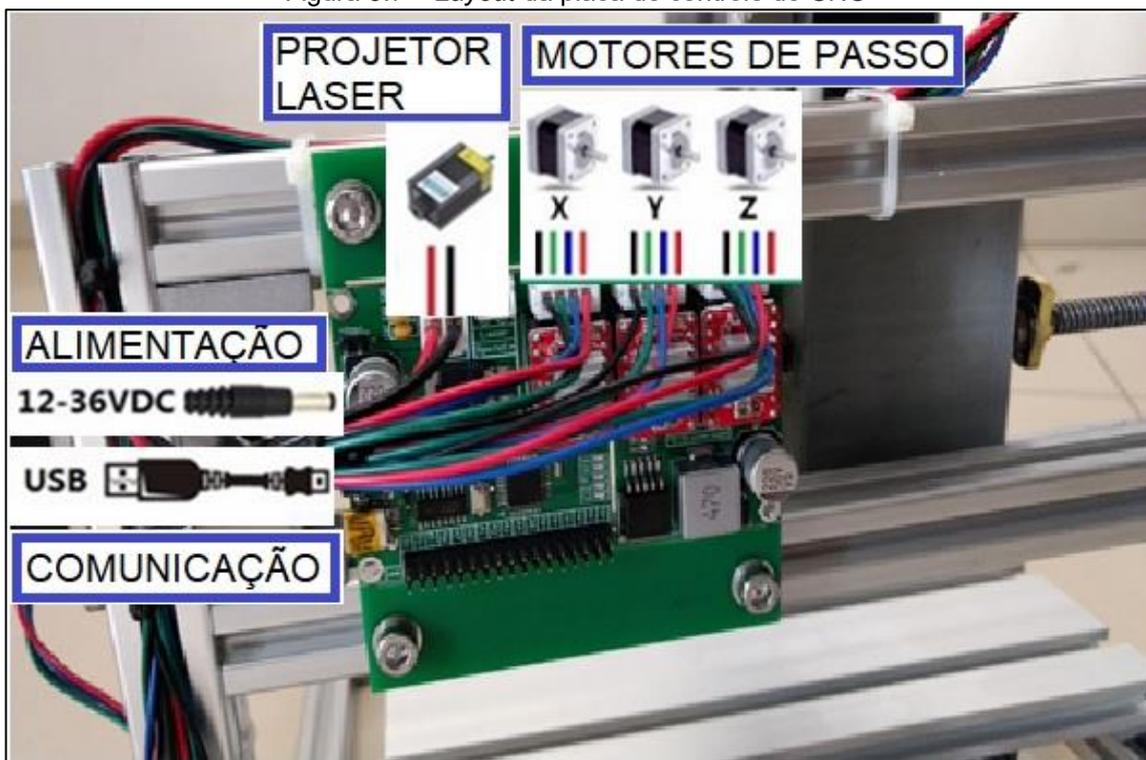
Fonte: Autoria própria, 2018

3.2 Instalação e configuração da placa de controle do CNC

O funcionamento da máquina CNC depende do uso de uma placa controladora do tipo GRBL. Ela é capaz de processar movimentos através dos eixos X, Y e Z. Sua conexão se faz através de porta USB e alimentação via fonte com tensões entre 12 a 36 V em corrente contínua.

A placa possui três controladores para os motores de passo com tensão de alimentação de 12 VDC e corrente máxima de 2 A. A alimentação para o Laser tem tensão de saída com 12 VDC e 5 A de corrente máxima. O *layout* da placa de controle com a disposição dos seus componentes é apresentado na Figura 3.7.

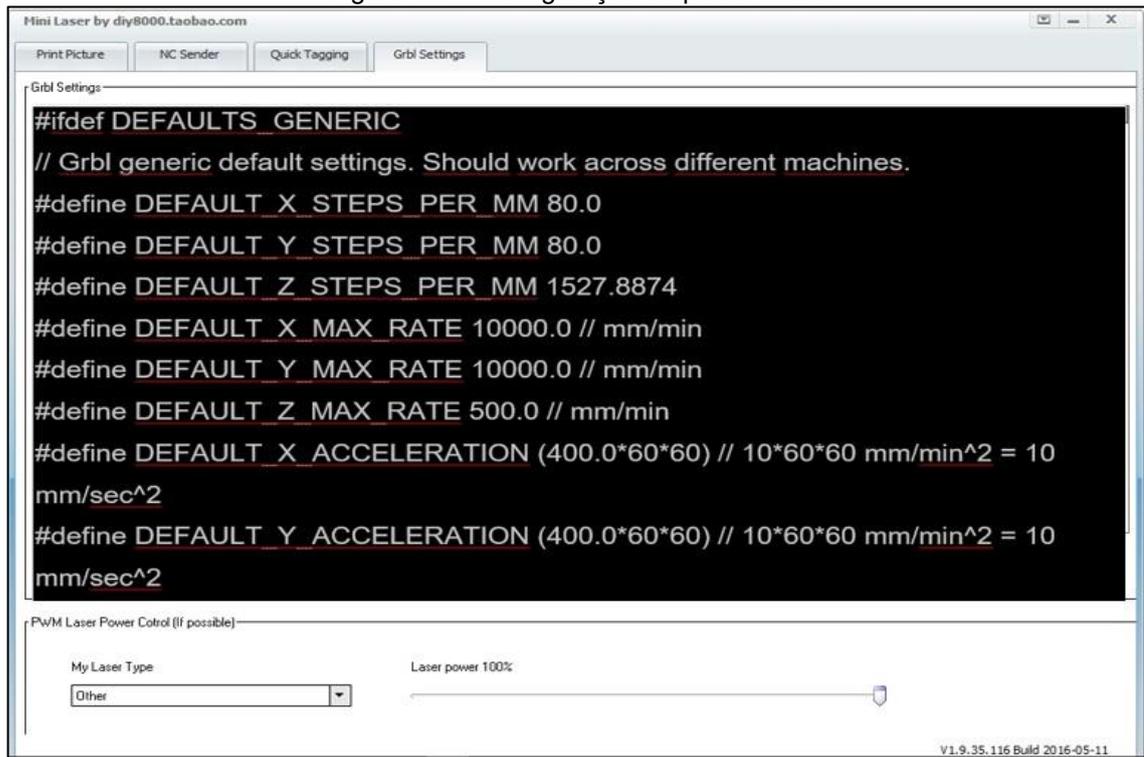
Figura 3.7 – *Layout* da placa de controle do CNC



Fonte: Foto de arquivo pessoal, 2018

Para configuração da placa GRBL a conexão é feita via cabo serial com taxa de transmissão de 115200 bits por segundo (bps) com 8 bits de dados, sem paridade e bit de 1 parada. Com a utilização da ferramenta Grbl Settings do *software* LiteFire a configuração é iniciada. A configuração da placa GRBL através do LiteFire é ilustrada na Figura 3.8.

Figura 3.8 – Configuração da placa GRBL



Fonte: Foto de arquivo pessoal, 2018

O primeiro comando para a impressão do menu de ajuda na tela é a tecla “\$” do teclado. A placa GRBL responde com uma lista das configurações atuais do sistema. Com todas as configurações feitas a GRBL responde com a mensagem “ok” confirmando que a placa está pronta para uso.

Todos os comandos para a configuração da placa GRBL se encontram no Anexo A. Os principais comandos para configuração da placa de controle são mostrados na Tabela 3.1.

Tabela 3.1 – Principais comandos de configuração da placa GRBL

| Comando | Função | Unidade |
|---------|---|----------------|
| \$ 0 | Pulso de passo | microssegundos |
| \$ 1 | Atraso de passo | microssegundos |
| \$ 2 | Inverte o sinal de pulso de passo | binário |
| \$ 3 | Inverte o sinal de direção para cada eixo | binário |

Fonte: Autoria própria, 2018

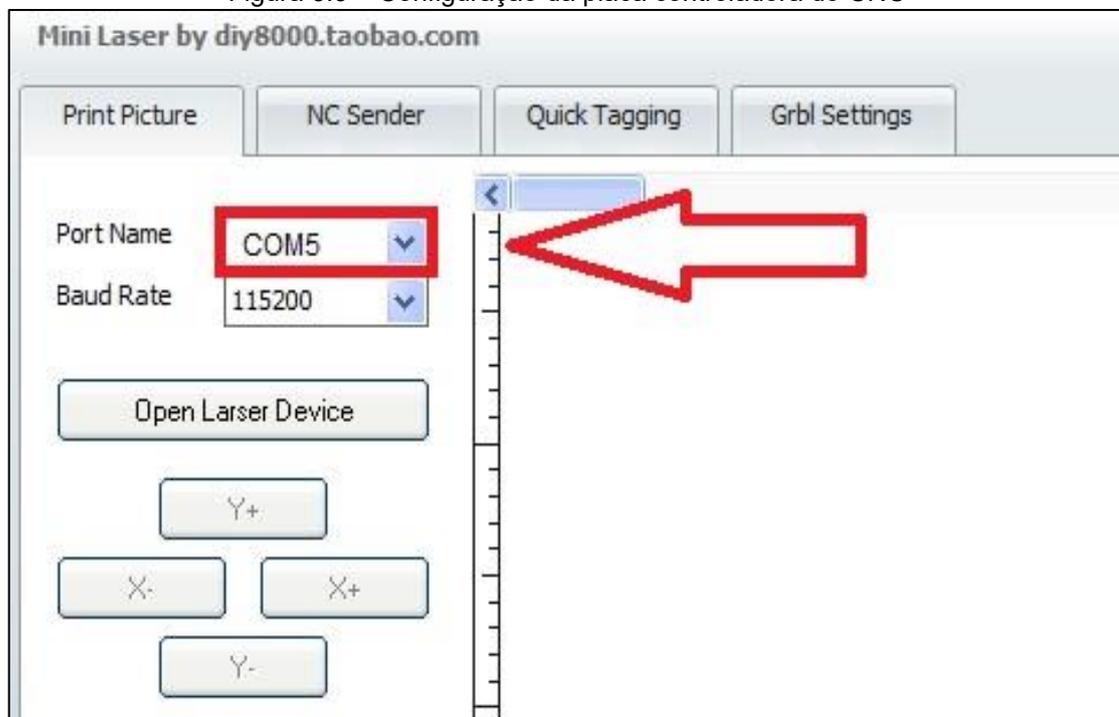
3.3 Teste preliminar de funcionamento da máquina CNC

O teste preliminar de funcionamento da máquina CNC é feito, inicialmente, com verificação eixo a eixo em manual para garantir que a máquina não possui erros de parametrização nem falsa leitura de fim de curso. Esses cuidados têm o intuito de encontrar alguma irregularidade mecânica durante o deslocamento dos componentes.

A placa controladora da máquina CNC utiliza o *software* LiteFire. O programa permite o uso da máquina no modo manual ou automático.

O programa LiteFire é um *software* de distribuição gratuita, porém com recursos adequados ao projeto. A área de trabalho da aplicação tem uma interface limpa e intuitiva apesar dos comandos em inglês. Com o programa aberto, é escolhido na guia configurações a porta COM5 como interface de comunicação entre o *software* e o controlador, conforme mostra a Figura 3.9.

Figura 3.9 – Configuração da placa controladora do CNC



Fonte: Foto de arquivo pessoal, 2018

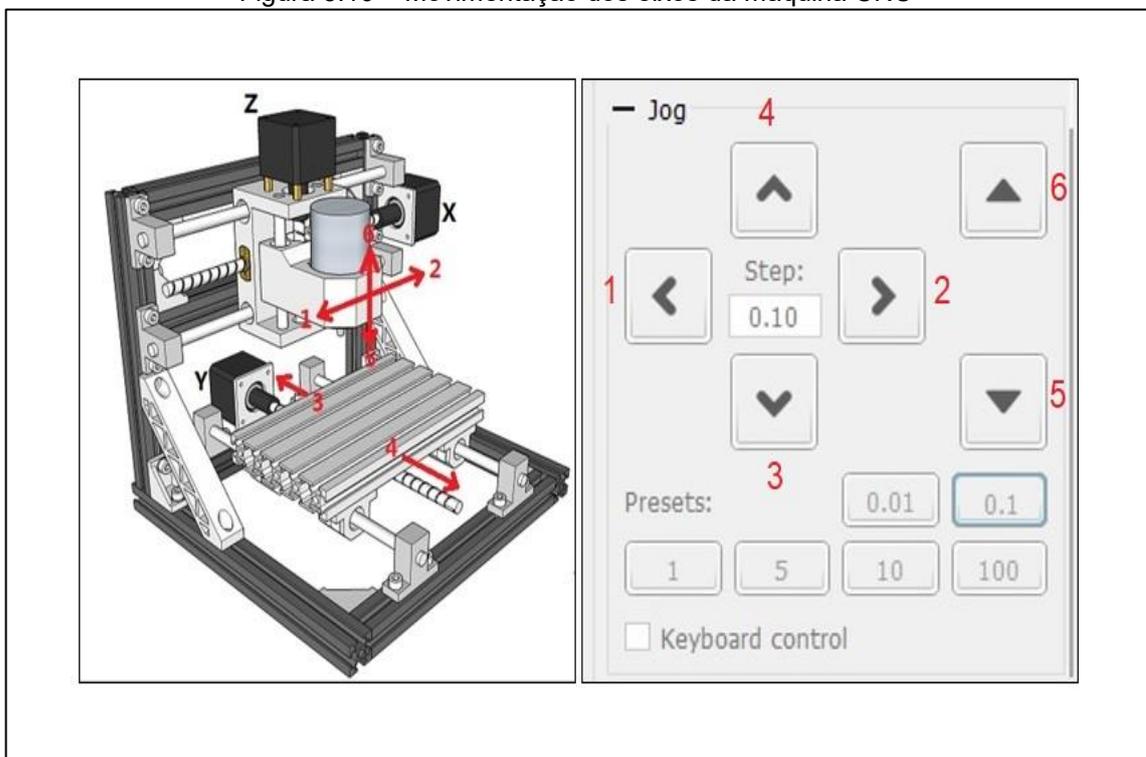
O teste se realiza da seguinte maneira: a máquina é ligada e a movimentação dos eixos ocorrem quando as respectivas teclas são pressionadas. Os movimentos

dos eixos são obtidos através da interface do *software* controlador da placa, selecionando:

- tecla 1 movimento do eixo em “X” para esquerda;
- tecla 2 movimento do eixo em “X” para direita;
- tecla 3 movimento do eixo em “Y” para trás;
- tecla 4 movimento do eixo em “Y” para frente;
- tecla 5 movimento do eixo em “Z” para baixo;
- tecla 6 movimento do eixo em “Z” para cima.

A movimentação dos eixos é verificada, conforme mostra a Figura 3.10.

Figura 3.10 – Movimentação dos eixos da máquina CNC



Fonte: Autoria própria, 2018

3.4 Desenvolvimento do supervisor com o *software* Elipse SCADA

Para que a automação do plano de manutenção seja efetivada, além da necessidade do uso da máquina CNC, para comprovar a possibilidade de se automatizar tal plano preventivo, se faz necessário uma plataforma que faça o trabalho de interface entre os sensores de vibração e os dados do plano de manutenção de onde são buscadas as ações para cada defeito ou falha que pode ocorrer no protótipo.

Como objeto de estudo e simplificação do projeto, a automação do plano de manutenção se dá apenas no controle de vibrações da máquina CNC. Com a confirmação da possibilidade do plano de manutenção através de sensores de vibrações, o projeto pode ser aprimorado com a inclusão de outros tipos de sensores.

O sistema de supervisão e aquisição de dados é construído através do *software* Elipse SCADA, que se mostra a ferramenta mais adequada para desempenhar o papel de monitor de vibração da máquina CNC. Com a construção do supervisor é feita a conexão entre a placa Arduino e o sensor de vibração.

A construção das páginas e *links* no Elipse SCADA são feitos de forma intuitiva e personalizada, porém a comunicação entre o supervisor e a placa Arduino requer a instalação de *drives*, que são comandos preestabelecidos, e programação específica para essa ação. O programa supervisor é composto de três telas: tela principal, telas Supervisão e tela Automação.

A tela principal é a primeira ser mostrada quando se roda a aplicação do supervisor de controle do sensor de vibração. Sua função é dar acesso as telas Supervisão e Automação através de botões ou de teclas de atalho no teclado do computador. A área de trabalho do supervisor pode ser verificada na Figura 3.11.

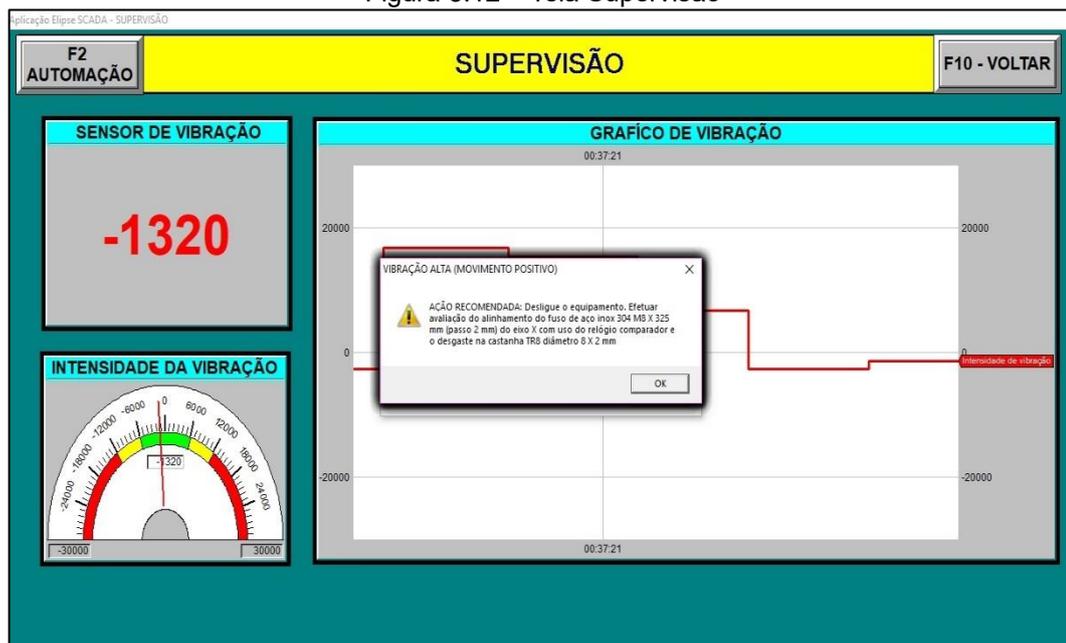
Figura 3.11 – Área de trabalho do supervisor



Fonte: Autoria própria, 2018

A tela Supervisão é composta de três tipos de ferramentas, sendo um *display* que mostra em tempo real o valor da vibração captada pelo sensor na máquina CNC em movimento, a outra é um *gauge* em forma de mostrador analógico com fundo de escala e agulha, que mostra os valores limites de intensidade de vibração e a última é um gráfico que ilustra a variação da vibração em um período, conforme ilustra a Figura 3.12.

Figura 3.12 – Tela Supervisão



Fonte: Autoria própria, 2018

A tela Automação mostra todas as falhas, onde são monitoradas por limites preestabelecidos. As vibrações que atingem os limites máximos aparecem na tela em forma de aviso. Eles informam o momento em que as falhas ocorrem, os valores de vibração atingidos e a ação recomendada para tratar a falha, conforme ilustra a Figura 3.13.

Figura 3.13 – Tela Automação

| dd/mm hh:mm:ss | Comentário | Valor | Limite |
|----------------|--|----------|--------|
| 08/05 00:37:21 | Vibração dentro dos padrões aceitáveis | 6724,0 | 0 |
| 08/05 00:37:19 | "VIBRAÇÃO ALTA" Avaliar alinhamento fuso 304 M8 X 325 mm "EIXO X" (MOVIMENTO POSITIVO) | 16808,0 | 12000 |
| 08/05 00:37:12 | Vibração dentro dos padrões aceitáveis | 2032,0 | 0 |
| 08/05 00:37:10 | "VIBRAÇÃO ALTA" Avaliar alinhamento fuso 304 M8 X 325 mm "EIXO X" (MOVIMENTO POSITIVO) | 16952,0 | 12000 |
| 06/05 13:47:59 | Vibração dentro dos padrões aceitáveis | 4452,0 | 0 |
| 06/05 13:47:58 | "VIBRAÇÃO ALTA" Avaliar alinhamento fuso 304 M8 X 325 mm "EIXO X" (MOVIMENTO POSITIVO) | 16180,0 | 12000 |
| 04/05 09:51:49 | Vibração dentro dos padrões aceitáveis | -816,0 | 0 |
| 04/05 09:51:49 | "VIBRAÇÃO ALTA" Avaliar alinhamento fuso 304 M8 X 325 mm "EIXO X" (MOVIMENTO NEGATIVO) | -12772,0 | -12000 |
| 04/05 09:39:51 | Vibração dentro dos padrões aceitáveis | 4468,0 | 0 |
| 04/05 09:39:08 | "VIBRAÇÃO ALTA" Avaliar alinhamento fuso 304 M8 X 325 mm "EIXO X" (MOVIMENTO POSITIVO) | 15048,0 | 12000 |
| 04/05 09:25:05 | Vibração dentro dos padrões aceitáveis | -1496,0 | 0 |
| 04/05 09:24:54 | "VIBRAÇÃO ALTA" Avaliar alinhamento fuso 304 M8 X 325 mm "EIXO X" (MOVIMENTO POSITIVO) | 15324,0 | 12000 |
| 04/05 09:24:15 | Vibração dentro dos padrões aceitáveis | 7276,0 | 0 |
| 04/05 09:24:12 | "VIBRAÇÃO ALTA" Avaliar alinhamento fuso 304 M8 X 325 mm "EIXO X" (MOVIMENTO POSITIVO) | 12960,0 | 12000 |
| 04/05 09:24:09 | Vibração dentro dos padrões aceitáveis | -9864,0 | 0 |
| 04/05 09:24:08 | "VIBRAÇÃO ALTA" Avaliar alinhamento fuso 304 M8 X 325 mm "EIXO X" (MOVIMENTO POSITIVO) | 16908,0 | 12000 |
| 03/05 12:45:58 | Vibração dentro dos padrões aceitáveis | -784,0 | 0 |
| 03/05 12:45:56 | "VIBRAÇÃO ALTA" Avaliar alinhamento fuso 304 M8 X 325 mm "EIXO X" (MOVIMENTO POSITIVO) | 12856,0 | 12000 |
| 03/05 12:45:53 | Vibração dentro dos padrões aceitáveis | 1300,0 | 0 |
| 03/05 12:45:52 | "VIBRAÇÃO ALTA" Avaliar alinhamento fuso 304 M8 X 325 mm "EIXO X" (MOVIMENTO NEGATIVO) | -15508,0 | -12000 |
| 03/05 12:44:32 | Vibração dentro dos padrões aceitáveis | -844,0 | 0 |
| 03/05 12:44:30 | "VIBRAÇÃO ALTA" Avaliar alinhamento fuso 304 M8 X 325 mm "EIXO X" (MOVIMENTO NEGATIVO) | -12884,0 | -12000 |

Fonte: Autoria própria, 2018

3.5 Configuração da placa Arduino controladora do supervisório

A placa escolhida para realizar a *interface* entre o *software* supervisório e o sensor de vibração no protótipo é a placa Arduino Mega2560. A placa possui 54 pinos de entradas e saídas digitais, onde 15 deles podem ser utilizadas como saídas PWM. Possui também 16 entradas analógicas e 4 portas de comunicação serial, além de possuir maior quantidade de pinos e de memória que as demais placas da plataforma Arduino. Possibilitando a criação de programas com maior capacidade.

Na comunicação da aplicação do Elipse SCADA com o uso do *software* Arduino é necessário instalar os *drivers* Modbus, para estabelecer a troca de dados mestre e escravo.

A programação é feita em linguagem C no IDE do Arduino, que é a área de programação do *software*. Inicialmente são incluídas as bibliotecas Modbusino.h e Wire.h que são especialmente criadas para o tipo de comunicação que a instalação do sensor de vibração pede. A Figura 3.14 mostra a programação inserida na placa Arduino de controle.

Figura 3.14 – Programação da placa Arduino



```

01_APL_ACL | Arduino 1.8.5
Arquivo | Editor | Sketch | Ferramentas | Ajuda
01_APL_ACL

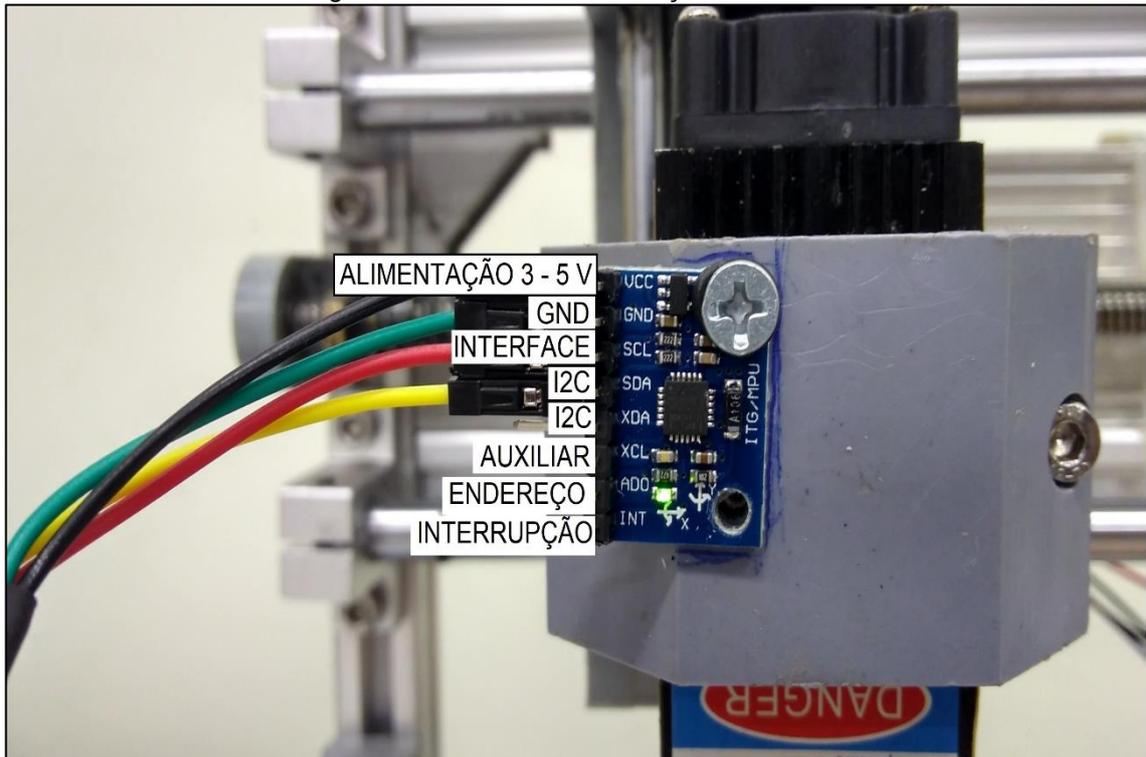
#include <Modbusino.h>
#include <Wire.h>
#define MPU1 0x68
#define MPU2 0x69
ModbusinoSlave modbusino_slave(1);
uint16_t tab_reg[10];
int AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
void setup() {
Wire.begin();
Wire.beginTransmission(MPU1);
Wire.write(0x6B);
modbusino_slave.setup(38400);
for(int i = 0; i < 10; i++) {tab_reg[i] = i;};
Wire.write(0);
Wire.endTransmission(true);
}
  
```

Fonte: Autoria própria, 2018

3.6 Comunicação do sensor de vibração com o supervisor

Dando prosseguimento, inicia-se a comunicação do sensor de vibração com o supervisor. O sensor de vibração utilizado é um módulo acelerômetro modelo MPU-6050, com *datasheet* que se encontra no Anexo B, possui boa sensibilidade na medição, que é facilmente captada durante as variações de aceleração em relação aos 3 eixos (x, y e z), a Figura 3.15 mostra o sensor de vibração MPU-6050.

Figura 3.15 – Sensor de vibração MPU-6050

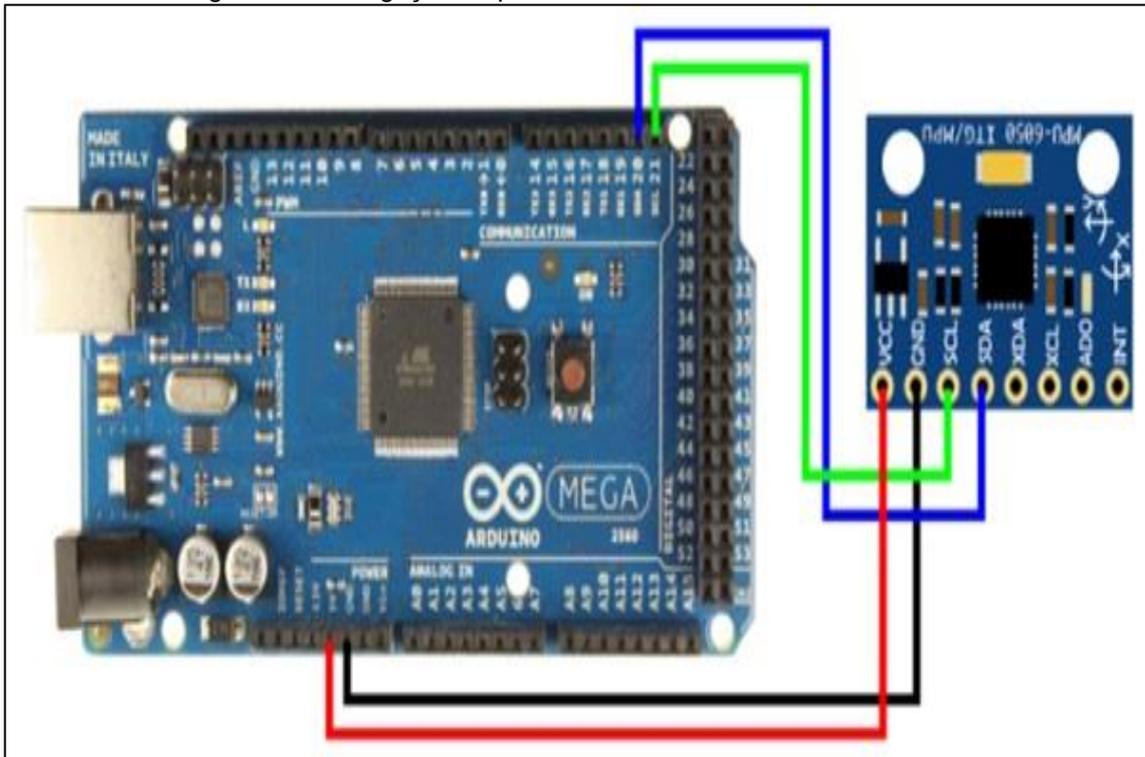


Fonte: Foto de arquivo pessoal, 2018

Para que o sensor colete dados é necessário criar um programa em linguagem C, adicionar as bibliotecas e declarar as portas de entrada e saída física que são usadas na placa Arduino em relação ao sensor de vibração. Essas informações dependem do meio físico e da grandeza que se registra. O programa criado para funcionamento do sensor encontra-se no Apêndice A.

O sensor de vibração é alimentado através do pino VCC com tensão de 3 V à 5 V possibilitando o uso de até dois sensores desse modelo diretamente ligados a placa Arduino. O pino GND é ligado ao pino terra da placa Arduino e os pinos SDA e SCL são ligados nas respectivas portas 20 (SDA) e 21 (SCL) de comunicação da placa Arduino. A Figura 3.16 mostra a ligação entre a placa Arduino e o sensor de vibração.

Figura 3.16 – Ligação da placa Arduino com o sensor MPU-6050



Fonte: Foto de arquivo pessoal, 2018

As informações são repassadas entre a placa e o sensor através da biblioteca *Wire*, que lê os dados e formata para enviar até o módulo receptor, permitindo a visualização da leitura do sensor conforme a aplicação criada no supervisório Elipse SCADA.

Para que ocorra a comunicação entre o Elipse SCADA e a placa Arduino é preciso usar o arquivo Modbus Master (*modbus.dll*). Dentro da aplicação do Elipse e no programa Arduino é compilado o arquivo Modbus slave.

O passo seguinte é entrar no atalho *Organizer* do programa Elipse SCADA e clicar no *link drive*. Isso permite selecionar o *drive* do Arduino com a versão compatível a comunicação e ao atrelamento das TAGs de aplicação, que são desenvolvidas para Elipse SCADA.

Além destas configurações, são especificados mais alguns parâmetros importantes relativos à comunicação a ser estabelecida, como a porta *COM* utilizada e a taxa de envio (*baud rate*) em bit por segundo, no caso do sensor de vibração para

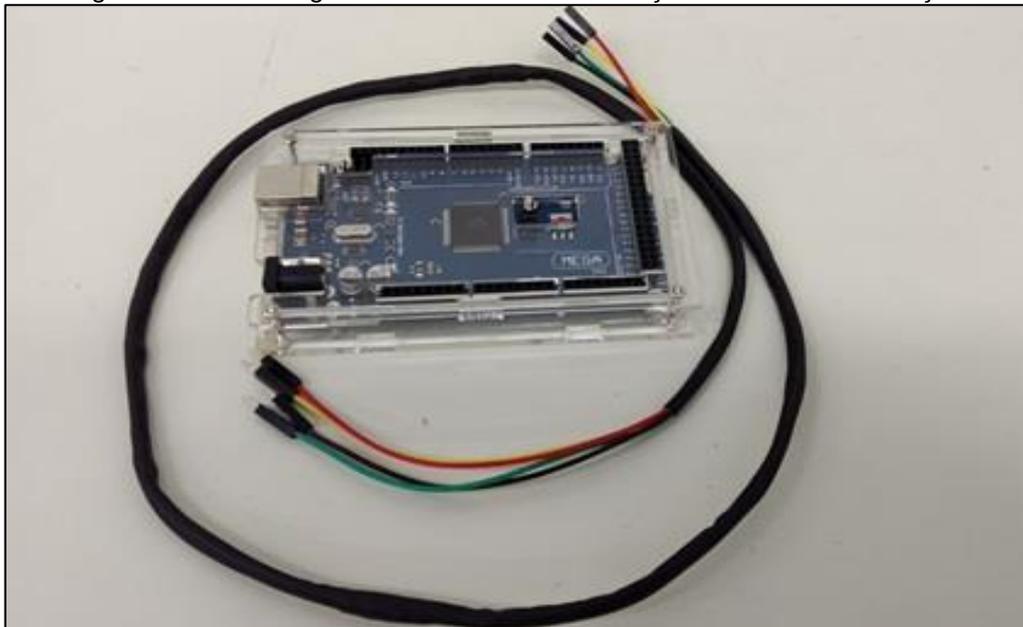
o melhor funcionamento é declarado no programa a velocidade 38400 bits por segundos (bps).

Para o sensor cumprir seu papel com eficiência é instalado no eixo da máquina mais sensível às vibrações, por isso, ele é instalado junto ao projetor laser captando movimentações dos eixos X e Z da máquina.

Os cabos de alimentação e comunicação do sensor de vibração, não podem interferir no sistema de movimentação da máquina CNC, por isso é montado um cabeamento com um comprimento de 400 mm, que é o tamanho adequado para se evitar colisão entre o sistema de movimentação da máquina é o cabeamento do módulo.

Nas emendas feitas nos cabos de alimentação e comunicação do sensor de vibração são utilizados material isolante do tipo plástico moldável, para melhorar o acabamento e a resistência dos cabos, como mostra a Figura 3.17.

Figura 3.17 – Montagem dos cabos de comunicação do sensor de vibração

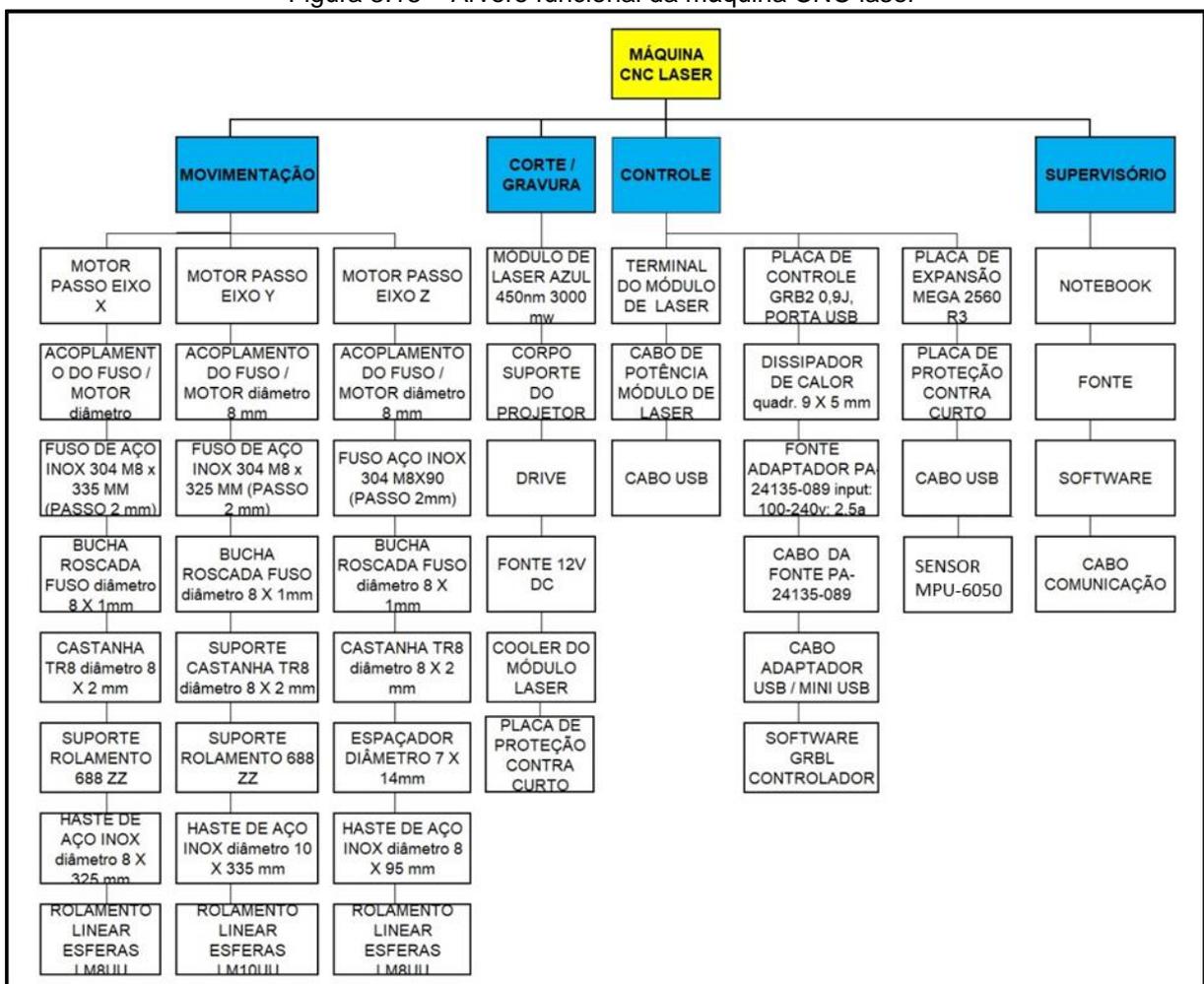


Fonte: A autoria própria, 2018.

3.7 Desenvolvimento e integração do plano de manutenção

O desenvolvimento do plano de manutenção foi baseado no uso da ferramenta FMEA, onde a máquina CNC é separada em sistemas de movimentação, corte / gravura, controle e supervisor. Dentro de cada sistema são analisadas as possíveis falhas e as ações necessárias para os principais componentes. Para facilitar a visualização da disposição dos sistemas e componentes envolvidos no projeto é criada a árvore funcional da máquina CNC laser, conforme ilustra a Figura 3.18.

Figura 3.18 – Árvore funcional da máquina CNC laser

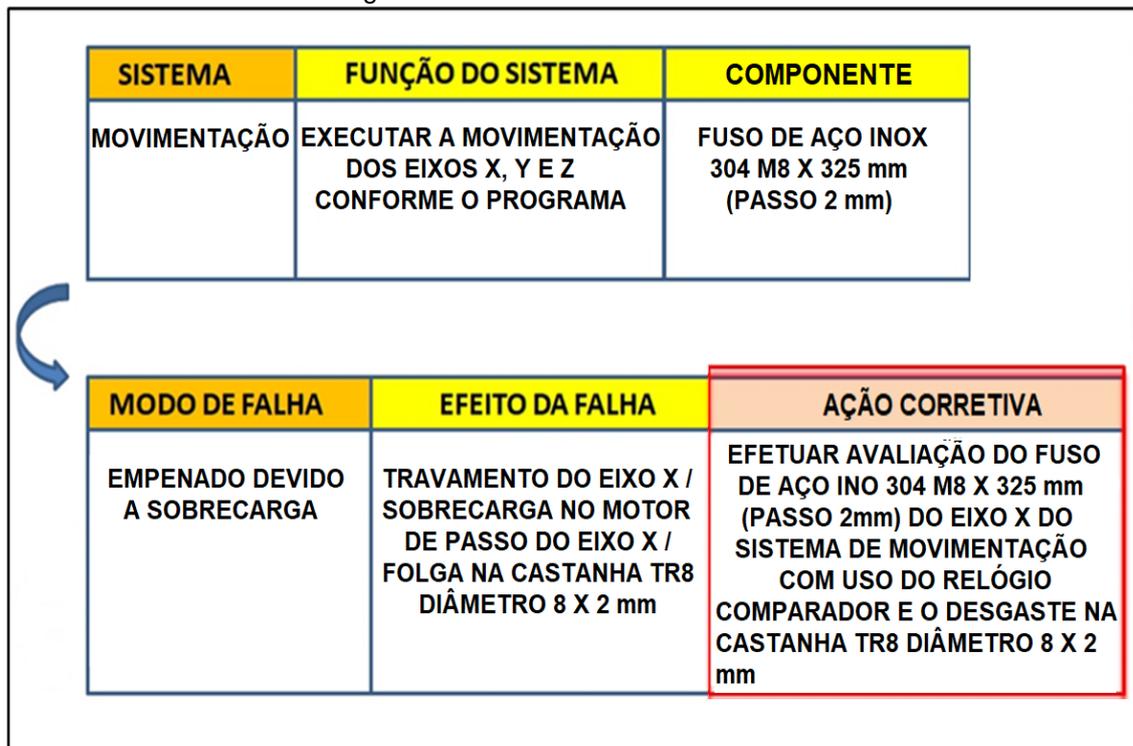


Fonte: Autoria própria, 2018.

Essas ações recomendadas ficam em um banco de dados de onde são extraídas informações para gerar uma mensagem na tela do supervisor conforme o valor preestabelecido nas configurações do *software* que informa qual a medida a ser

tomada diante de cada tipo de falha. A Figura 3.19 exemplifica o funcionamento da FMEA no plano de manutenção com a ação corretiva usada no projeto.

Figura 3.19 – Funcionamento da FMEA



Fonte: Autoria própria, 2018.

A supervisão do processo de controle de vibração ocorre através da leitura de variáveis que são captadas pelo sensor, onde os valores dessas variáveis são associados a objetos do sistema chamados TAGs. Todas as variáveis são representadas de forma numéricas ou alfanuméricas, conforme o tipo de aplicação e uso.

3.8 Testes finais e funcionamento do projeto

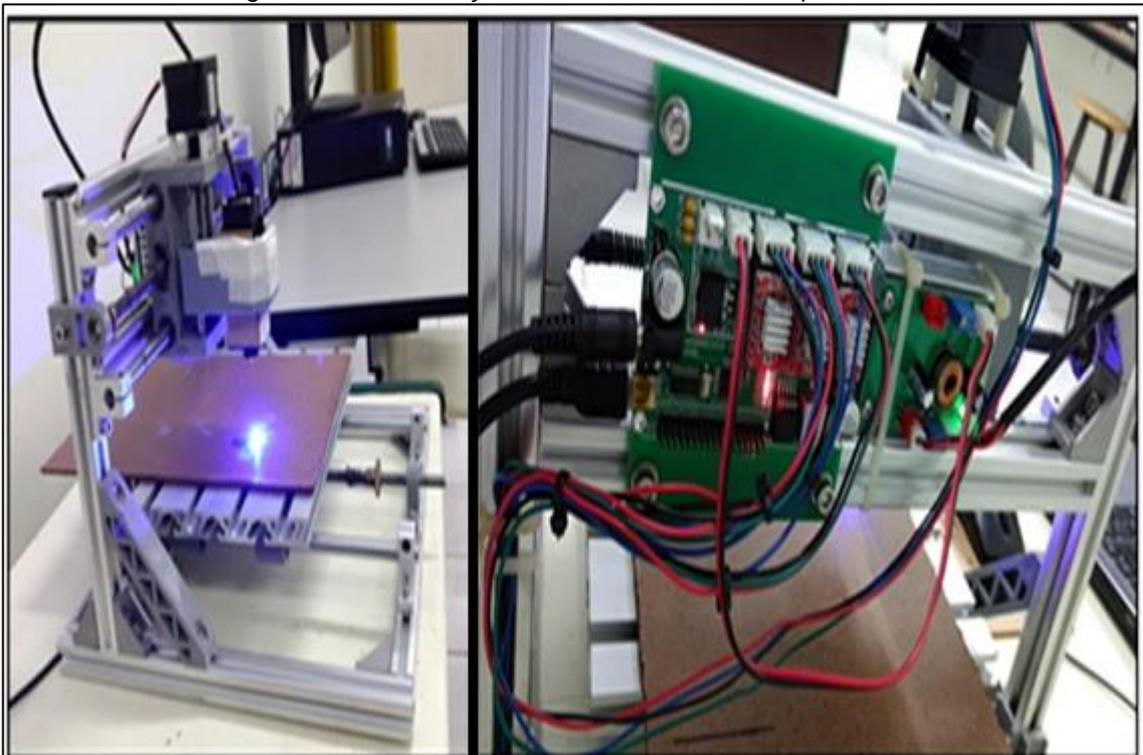
Concluídos o teste preliminar de funcionamento do protótipo é dado início a fase de teste de funcionamento da máquina, para verificar se os requisitos desejados e as metas foram alcançadas.

No teste preliminar é aferida e fixada a velocidade de avanço adequadas à gravação das imagens, sem que ocorram trombadas ou falha na impressão. Esse

ajuste garante que quando instalado o projetor laser a máquina está pronta para operar sem maiores ajustes.

Com a substituição do *spindle* pelo módulo laser o equipamento é fixado no mesmo suporte do eixo Z da máquina CNC. Em seguida é ligado à sua fonte de 12 VDC na entrada específica da placa GRBL, o cabo de comunicação USB vai da placa ao computador, conforme mostra a Figura 3.20.

Figura 3.20 – Instalação do módulo laser na máquina CNC



Fonte: Autoria própria, 2018.

Com a instalação do laser no topo da máquina e com os ajustes de distância, foco e potência é necessária uma última etapa para poder realizar seu funcionamento. Esta etapa é a instalação de um *software* para controlar o projetor e permitir o seu acionamento, controle e escolha das imagens.

O *software* que melhor realiza essas tarefas é o LiteFire. O programa tem distribuição gratuita e fácil operação, motivos que ajudam na contenção de custos do projeto. Ele roda sem a necessidade de instalação, é pequeno e consome pouco espaço de memória do computador. Para iniciar o *software* basta clicar em seu logotipo na área de trabalho. Neste momento um arquivo de registro de sistema (log)

é gerado permitindo o registro das principais funções do computador que está rodando a aplicação.

Para a criação de uma imagem ou de um texto e projeção do laser na peça o aplicativo permite a transformação de uma figura ou texto em uma linguagem de máquina, ou seja, em código G, que é carregada pelo próprio programa para enviar ao CNC. A interface do programa e seus comandos podem ser vistos na Figura 3.21.

Figura 3.21 – Interface do *software* LiteFire



Fonte: Foto de arquivo pessoal, 2018.

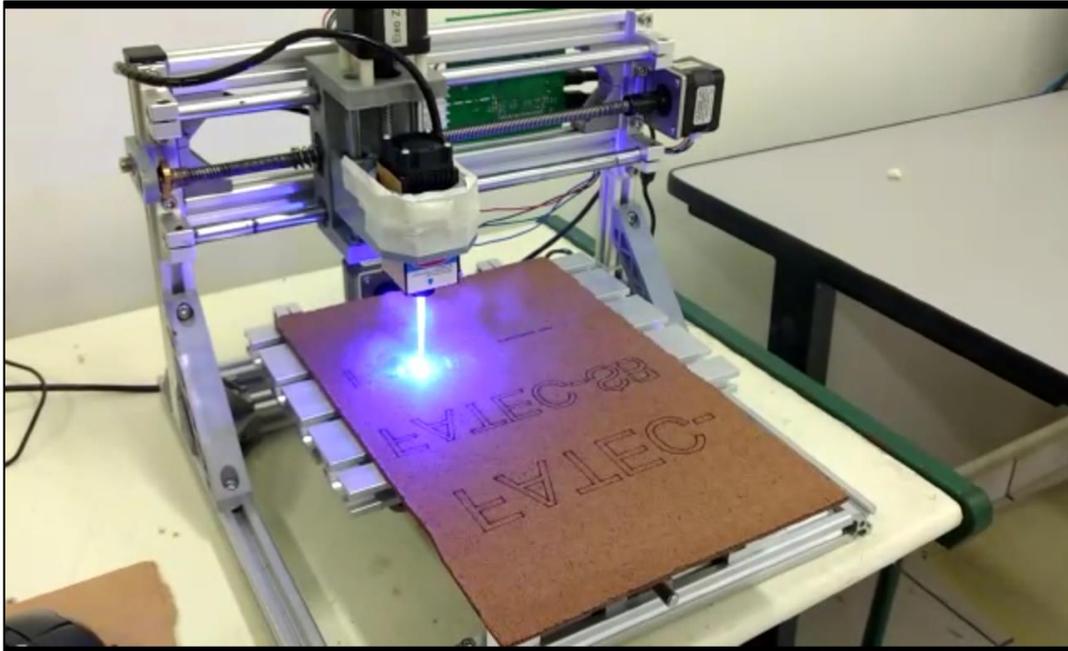
Com a instalação do projetor laser e do *software* controlador é dado início aos testes de funcionamento. A imagem para a realização do teste pode ser do arquivo pré-existente do *software* controlador, criada na área de trabalho do *software* ou carregada de um arquivo pessoal.

Através da tecla *Load Picture* a busca da imagem é feita e inserida na área de trabalho. Para que essa imagem seja projetada na peça através do laser, o comando no programa é a tecla *Start*, mas antes de enviar este comando é necessário o ajuste prévio do foco do laser diretamente no projetor girando o suporte da lente no sentido em que deixa o ponto do feixe de luz mais nítido na placa de teste. Isso garante a qualidade na definição do acabamento e a precisão no ponto realizado. Por medida

de segurança essa tarefa é executada com o equipamento sem nenhuma fonte de alimentação.

Com todos os requisitos técnicos e de segurança atendido a máquina imprime a imagem do teste sem erros e com boa definição, conforme mostra a Figura 3.22.

Figura 3.22 – Impressão de imagem de teste



Fonte: Autoria própria, 2018.

Concluídos testes finais é chegado o momento de verificar se todos os requisitos na construção do protótipo foram atendidos. O escopo inicial do projeto prevê a construção de uma máquina capaz de poder receber um sistema de sensoriamento que envia informações do seu plano de manutenção.

Com o banco de dados disponível a ferramenta monitora em tempo real o equipamento e orienta o setor de manutenção nas melhores práticas de controle, conservação e operação dos seus equipamentos e maquinas em uma unidade fabril.

Os resultados esperados de tal ferramenta são a racionalização dos recursos, maior tempo de vida útil do equipamento e menor tempo do equipamento para reparo e revisão.

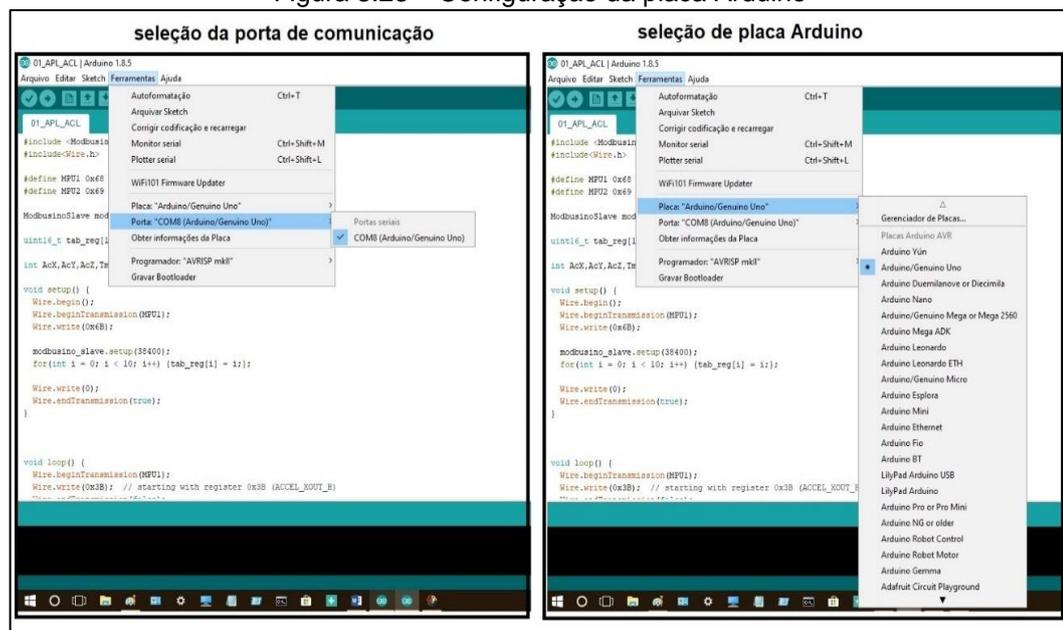
Para o perfeito funcionamento do projeto deve-se ter obrigatoriamente dois computadores. Um para rodar o programa supervisor que vai monitorar a máquina CNC através do sensor de vibração e definir as ações do plano de manutenção que serão visualizadas.

O outro computador vai fazer o controle do CNC através do *software* LiteFire. Outro dispositivo importante para o funcionamento do projeto é a placa Arduino, através da placa os comandos em seu interior dão os parâmetros necessários para o funcionamento do sensor de vibração.

O computador 1 com programa supervisor é ligado e aberto o *software* Arduino. Sua função é compilar (enviar) para dentro da placa Arduino os comandos em linguagem C para funcionamento do sensor de vibração. O envio se realiza via cabo USB. Assim que aberto o Arduino os principais parâmetros configurados são, na aba ferramentas a escolha de placa a ser usada e a porta de comunicação.

Através da aba abrir o programa já criado é aberto. Para o envio do programa para a placa Arduino antes se faz necessário uma compilação, que é a verificação se todos os dados do programa estão corretos. Após compilado o programa é carregado na placa. Essa configuração é ilustrada na Figura 3.23.

Figura 3.23 – Configuração da placa Arduino



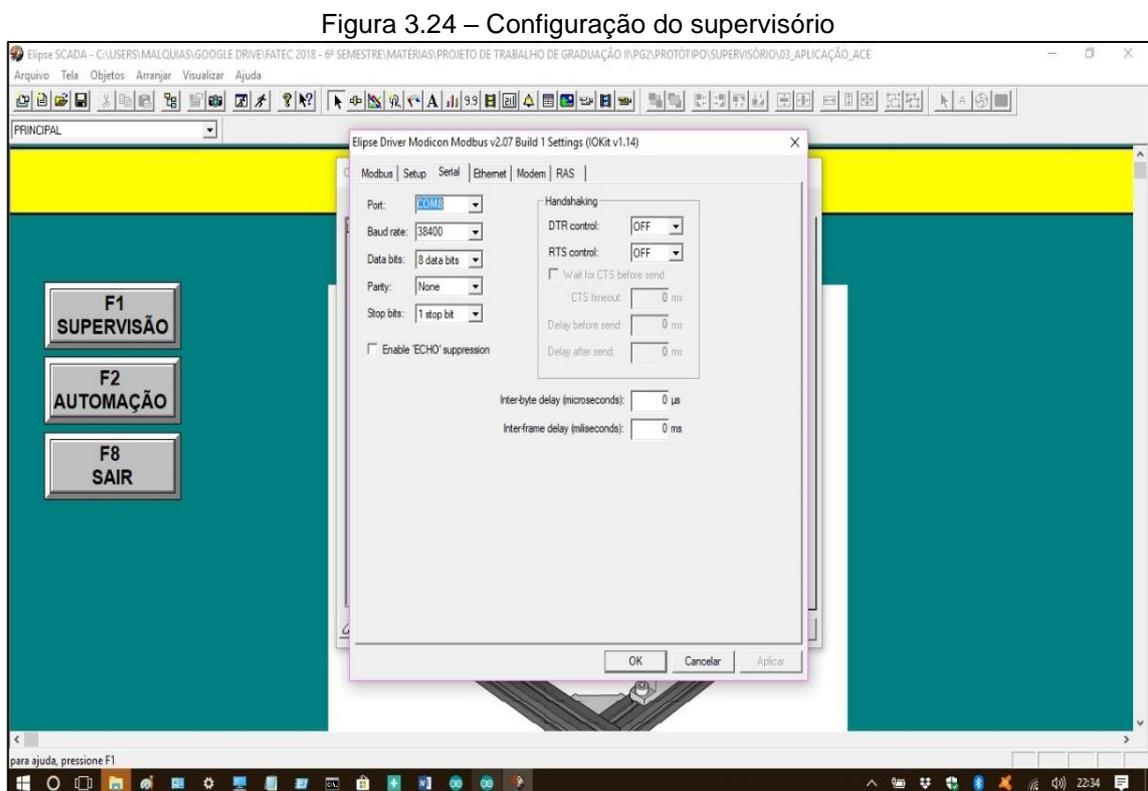
Fonte: Autoria própria, 2018.

Uma pré-visualização dos dados recebidos pelo sensor são vistos na aba monitor do Arduino. Esse procedimento de abertura do Arduino, escolha da placa a ser usada e porta de comunicação tem que ser realizado apenas uma vez.

Assim que carregado o programa na placa, o *software* é fechado e só precisa ser aberto novamente se alguma edição no programa, troca de placa ou troca de porta serial se faz necessários.

Após a configuração da placa Arduino, o programa Elipse SCADA é aberto, na aba abrir é escolhido o programa supervisorio no computador que supervisiona o sensor de vibração. Assim que aberto o supervisorio deve ser configurado ainda em seu modo edição (antes de rodar).

Na aba *organize* dentro da opção drive, a porta de comunicação tem que ser inserida, conforme mostra a Figura 3.24.



Fonte: Autoria própria, 2018.

Feito essa configuração a aplicação pode ser iniciada. Neste momento o supervisorio já está captando todos os dados enviados do CNC pelo sensor de

vibração. Como o CNC ainda não está funcionando os dados enviados têm valores próximos de zero.

O computador 2 opera o CNC através do *software* LiteFire. Como todos os parâmetros do CNC já foram gravados nos testes preliminar e finais, nenhuma ação de configuração se faz necessária. Basta escolher a imagem arquivada em algum diretório do computador ou criar uma na hora.

Assim que inserida imagem na área de trabalho do LiteFire o CNC está pronto para operar. É dado o comando de *start* e a máquina começa a emitir luz e seguir as linhas e traços do desenho. Construindo a imagem pelo processo de impressão por aquecimento. Enquanto o CNC trabalha os dados de vibração aparecem na tela do supervisor do computador 1. Um dos motivos da escolha de uma máquina CNC é que esse tipo de equipamento é extremamente sensível a vibração, tornando-a ideal como objeto de estudo ao projeto. Como o CNC é uma máquina nova e recém montada a chance de registro de qualquer anomalia no seu funcionamento é pequena, por isso os índices de vibração foram manipulados para deixar o sensor no seu maior grau de sensibilidade.

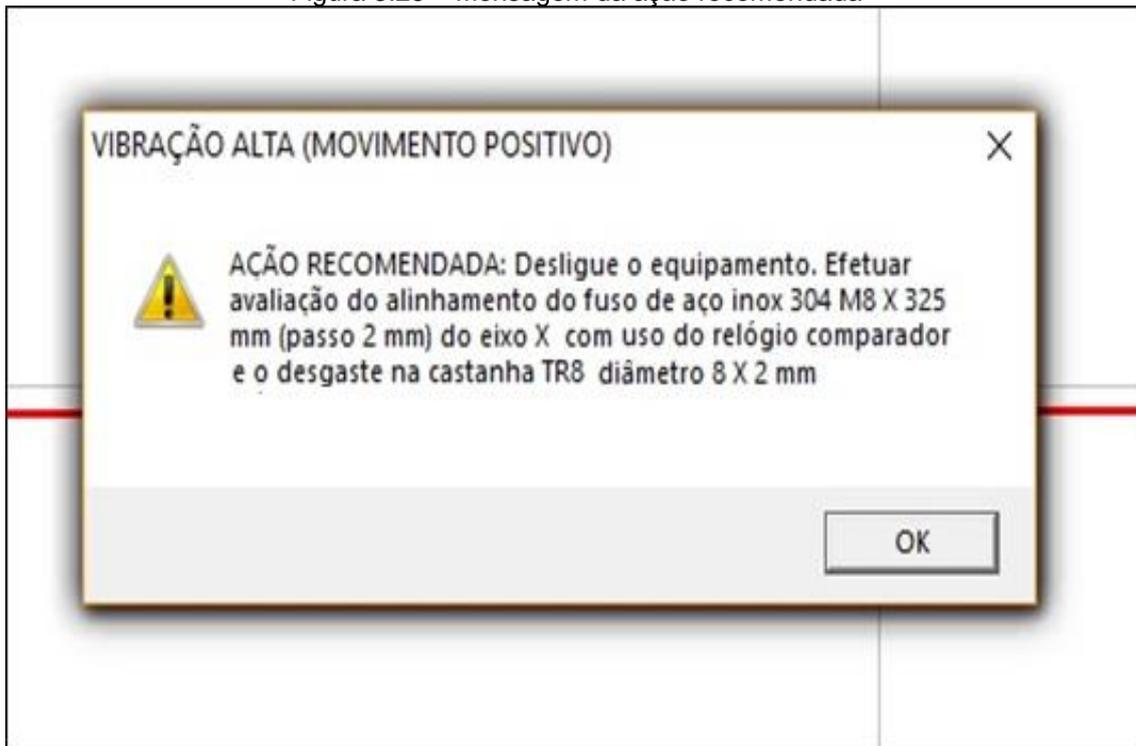
Os parâmetros de sensibilidade dos alarmes são modificados também no software supervisor. Os principais alarmes são, vibração alta valor de referência 12.000 (movimento positivo). O movimento negativo tem vibração alta valor de referência -12.000. Os valores mostrados pelo acelerômetro não têm uma unidade pois são apenas valores de referência, que têm a função de acionar as ações recomendadas pelo plano de manutenção.

A ações recomendadas do plano de manutenção acionadas pelos alarmes para movimento positivo ou negativo são:

- Desligue o equipamento. Efetuar avaliação do alinhamento do fuso de aço inox 304 M8 X 325 mm (passo 2 mm) do eixo X da máquina CNC laser com uso do relógio comparador e o desgaste na castanha TR8 diâmetro 8 X 2 mm", "VIBRAÇÃO ALTA"

Com o acelerômetro e o supervisor ajustados para a menor sensibilidade de vibração emitida pelo CNC. O projeto mostra toda sua funcionalidade. Todas as vibrações são captadas e em tempo real as ações recomendadas pelo plano de manutenção são enviadas ao operador do supervisor de acordo ilustra a Figura 3.25.

Figura 3.25 – Mensagem da ação recomendada



Fonte: Autoria própria, 2018.

3.9 Obstáculos e soluções durante o desenvolvimento do projeto

Durante o desenvolvimento e construção do projeto deparou-se com alguns obstáculos, que foram solucionados através de conhecimentos adquiridos nas pesquisas, experiência profissional dos integrantes do grupo e consulta a professores.

Obstáculo 1: dificuldade no sensoriamento e controle da máquina, pois grande parte dos sensores testados não conseguiam captar os dados de vibração no equipamento.

Solução: implantação do sensor de vibração na máquina CNC. Ele é capaz de captar as menores vibrações em todos os sentidos. Sensor ideal para os requisitos do projeto.

Obstáculo 2: ao interligar o sensor de vibração com a placa Arduino e o supervisor não existia um programa de comunicação.

Solução: buscou-se em sites e artigos na internet exemplos de comunicação entre os dispositivos.

CONSIDERAÇÕES FINAIS

O objetivo desse trabalho que se intitula Automação do plano de manutenção da máquina CNC laser é montar uma máquina CNC de corte a laser para criar um plano de manutenção preventiva voltado a ela. Justifica-se por maior acessibilidade de manutenção, facilidade na interação homem-máquina (IHM), melhoria na supervisão de falhas através de telas de supervisórios, permissão do controle local das atividades em tempo real, execução de gravuras e cortes a laser, minimização de custo, maior controle de manutenção, menor geração de resíduos e consequentemente uma redução no impacto ambiental.

Para o desenvolvimento do projeto fez-se uso de um sensor acelerômetro que tem por finalidade captar as vibrações durante o deslocamento do eixo X da máquina CNC. A monitoração e controle dos dados é enviado para uma placa de controle onde as informações são processadas conforme programação e transmitidos através de um *software* que faz a função de uma interface homem-máquina informando em tempo real a situação local do componente e ou conjunto.

Também, para a construção do projeto foi montado o CNC, fez-se uso software Elipse Scada, software LiteFire, placa de controle Arduino, placa GRBL do módulo CNC e o projetor laser e dois notebooks.

As teorias pesquisadas foram de suma importância para os integrantes do grupo conhecerem melhor o funcionamento da automação de um plano de manutenção. O caminho percorrido para a execução do trabalho mostrado pela metodologia científica é essencial para a organização, direcionamento e suporte dos objetivos propostos. Nela encontram-se os métodos e técnicas que direcionam o desenvolvimento do projeto.

Como vantagens destacam-se o aumento da disponibilidade da máquina, na redução dos custos com estoque de peças, na padronização das ações recomendadas do plano de manutenção e principalmente para os integrantes do grupo que obtiveram uma visão mais ampla do uso da automação, onde é possível

aplicá-la em atividades relacionadas a outros ramos. Tem como desvantagens a necessidade de pequenos ajustes para implantá-lo na indústria.

Durante o desenvolvimento do projeto deparou-se com alguns transtornos, como a dificuldade no sensoriamento e controle da máquina, pois grande parte dos sensores testados não conseguiam captar os dados de vibração no equipamento, ao interligar o sensor de vibração com a placa Arduino e o supervisor não existia um programa de comunicação. Estes obstáculos foram solucionados com as teorias pesquisadas e conhecimentos baseados nas experiências dos integrantes.

Como sugestão para futuros trabalhos relacionados ao tema-problema a substituição do uso da placa Arduino Mega2560 por um C.L.P (Controlador Lógico Programável). Também pode ser melhorado com a migração do software para uso do seu armazenamento em nuvem, permitindo que o controle dos sistema e arquivos ocorra de forma remota, proporcionando maior mobilidade, redução nos custos, centralização de suporte a manutenção e maior disponibilidade.

REFERÊNCIAS

- CASSANIGA, F. A. **Fácil programação do controle numérico**. 2. ed. Joinville: Clube dos Autores, 2005.
- CIÊNCIA HOJE. **O magnífico laser**. Disponível em: <http://www.cienciahoje.org.br/revista/materia/id/128/n/o_magnifico_laser>. Acesso em: 12 set. 2017.
- DALOSTO, Diogo Nunes; BALESTRASSI, Pedro Paulo; PERUCHI Rogério S. **Análise da consistência do FMEA: uma abordagem quantitativa**. 1. ed. São Paulo: Novas Edições Acadêmicas, 2015.
- ESPINOLA, Mauro; PESSOA, Marcelo. **Introdução à automação**. 1. ed. Rio de Janeiro: Elsevier, 2014.
- EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino em ação**. 1. ed. São Paulo: Novatec, 2013.
- FITZPATRICK, Michael. **Introdução à usinagem com CNC: Série Tekne**. 1. ed. Curitiba: Bookman, 2013.
- FOGLIATO, Flavio; RIBEIRO, José. **Confiabilidade e manutenção industrial**. 1. ed. Rio de Janeiro: Elsevier, 2009
- MANUAL DE NORMALIZAÇÃO DE PROJETO DE TRABALHO DE GRADUAÇÃO – FATEC SÃO BERNARDO DO CAMPO. **Material didático para utilização nos projetos de trabalho de graduação dos cursos de tecnologia em automação industrial e informática**. São Bernardo do Campo: Fatec, 2017.
- MCROBERTS, M. **Arduino básico**. 2. ed. São Paulo: Novatec, 2015.
- MONK, Simon. **Programação com Arduino: Começando com Sketches**. 2. ed. Curitiba: Bookman, 2017.
- PALADY, Paul. **FMEA: análise dos modos de falha e efeitos: prevendo e prevenindo problemas antes que ocorram**. 1. ed. São Paulo: IMAM, 2004.
- PREECE, Jennifer; ROGERS, Yvonne; SHARP, Helen. **Design de interação: Além da interação homem-computador**. 1. ed. Porto Alegre: Bookman, 2005.
- PRODANOV, C. C.; FREITAS, E. C. **Metodologia do trabalho científico: Métodos e técnicas da pesquisa e do trabalho científico**. 2. ed. Rio Grande do Sul: Universidade Feevale, 2013.
- ROSÁRIO, João M. **Automação industrial**. 1. ed. São Paulo: Baraúna, 2009.
- SEARA DA CIÊNCIA. **Lasers**. Disponível em: <<http://www.searadaciencia.ufc.br/especiais/fisica/lasers/lasers.htm>>. Acesso em: 01 set. 2017.

SEVERINO, A. J. **Metodologia do trabalho científico**, 24. ed. São Paulo: Cortez, 2017.

SILVA, S. D. **CNC programação de comandos numéricos computadorizados**. 4. ed. São Paulo: Érica, 2015.

ZIEMANN, Volker. **Um curso prático de sensores usando o Arduino e o Raspberry Pi**. Nova York: CRC Press, 2018.

APÊNDICE A – CÓDIGOS DE PROGRAMAÇÃO DA PLACA ARDUINO

```

#include <Modbusino.h>
#include<Wire.h>
#define MPU1 0x68
#define MPU2 0x69

ModbusinoSlave modbusino_slave(1);

uint16_t tab_reg[10];

int AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

void setup() {
  Wire.begin();
  Wire.beginTransmission(MPU1);
  Wire.write(0x6B);

  modbusino_slave.setup(38400);
  for(int i = 0; i < 10; i++) {tab_reg[i] = i;}

  Wire.write(0);
  Wire.endTransmission(true);
}
void loop() {
  Wire.beginTransmission(MPU1);
  Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);

  //Solicita os dados do sensor
  Wire.requestFrom(MPU1,14,true);

  //Armazena o valor dos sensores nas variaveis correspondentes
  AcX= Wire.read()<<8|Wire.read(); //0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY=Wire.read()<<8|Wire.read(); //0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ=Wire.read()<<8|Wire.read(); //0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Tmp=Wire.read()<<8|Wire.read(); //0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
  GyX=Wire.read()<<8|Wire.read(); //0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
  GyY=Wire.read()<<8|Wire.read(); //0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
  GyZ=Wire.read()<<8|Wire.read(); //0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)

  //Mostra os valores na serial
  Serial.print("Acel. X = "); Serial.print(AcX);
  Serial.print(" | Y = "); Serial.print(AcY);
  Serial.print(" | Z = "); Serial.print(AcZ);
  Serial.print(" | Gir. X = "); Serial.print(GyX);
  Serial.print(" | Y = "); Serial.print(GyY);
  Serial.print(" | Z = "); Serial.print(GyZ);
  Serial.print(" | Temp = "); Serial.println(Tmp/340.00+36.53);

  modbusino_slave.loop(tab_reg, 10);
}

```

ANEXO A – COMANDOS PARA CONFIGURAÇÃO DA PLACA GRBL

```
#ifndef DEFAULTS_GENERIC
// Grbl generic default settings. Should work across different machines.
#define DEFAULT_X_STEPS_PER_MM 80.0
#define DEFAULT_Y_STEPS_PER_MM 80.0
#define DEFAULT_Z_STEPS_PER_MM 1527.8874
#define DEFAULT_X_MAX_RATE 10000.0 // mm/min
#define DEFAULT_Y_MAX_RATE 10000.0 // mm/min
#define DEFAULT_Z_MAX_RATE 500.0 // mm/min
#define DEFAULT_X_ACCELERATION (400.0*60*60) // 10*60*60 mm/min^2 = 10
mm/sec^2
#define DEFAULT_Y_ACCELERATION (400.0*60*60) // 10*60*60 mm/min^2 = 10
mm/sec^2
#define DEFAULT_Z_ACCELERATION (100.0*60*60) // 10*60*60 mm/min^2 = 10
mm/sec^2
#define DEFAULT_X_MAX_TRAVEL 150.0 // mm
#define DEFAULT_Y_MAX_TRAVEL 200.0 // mm
#define DEFAULT_Z_MAX_TRAVEL 22.0 // mm
#define DEFAULT_STEP_PULSE_MICROSECONDS 15
#define DEFAULT_STEPPING_INVERT_MASK 0
#define DEFAULT_DIRECTION_INVERT_MASK 1
#define DEFAULT_STEPPER_IDLE_LOCK_TIME 255 // msec (0-254, 255 keeps
steppers enabled)
#define DEFAULT_STATUS_REPORT_MASK 3
#define DEFAULT_JUNCTION_DEVIATION 0.02 // mm
#define DEFAULT_ARC_TOLERANCE 0.002 // mm
#define DEFAULT_REPORT_INCHES 0 // false
#define DEFAULT_AUTO_START 1 // true
#define DEFAULT_INVERT_ST_ENABLE 0 // false
#define DEFAULT_INVERT_LIMIT_PINS 0 // false
#define DEFAULT_SOFT_LIMIT_ENABLE 0 // false
#define DEFAULT_HARD_LIMIT_ENABLE 0 // false
#define DEFAULT_HOMING_ENABLE 1 // false
```

```
#define DEFAULT_HOMING_DIR_MASK 0 // move positive dir
#define DEFAULT_HOMING_FEED_RATE 2500.0 // mm/min
#define DEFAULT_HOMING_SEEK_RATE 2500.0 // mm/min
#define DEFAULT_HOMING_DEBOUNCE_DELAY 150 // msec (0-65k)
#define DEFAULT_HOMING_PULLOFF -1.0 // mm
#endif
```

```
$0=10 (step pulse, usec)
$1=25 (step idle delay, msec)
$2=0 (step port invert mask:00000000)
$3=0 (dir port invert mask:00000000)
$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=0 (homing dir invert mask:00000000)
$24=25.000 (homing feed, mm/min)
$25=500.000 (homing seek, mm/min)
$26=250 (homing debounce, msec)
$27=1.000 (homing pull-off, mm)
$100=250.000 (x, step/mm)
$101=250.000 (y, step/mm)
$102=250.000 (z, step/mm)
$110=500.000 (x max rate, mm/min)
$111=500.000 (y max rate, mm/min)
$112=500.000 (z max rate, mm/min)
$120=10.000 (x accel, mm/sec^2)
```

\$121=10.000 (y accel, mm/sec²)

\$122=10.000 (z accel, mm/sec²)

\$130=200.000 (x max travel, mm)

\$131=200.000 (y max travel, mm)

\$132=200.000 (z max travel, mm)

ANEXO B – DATASHEET DO SENSOR DE VIBRAÇÃO MPU6050

Electrical Characteristics Gyroscope Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, TA = 25°C

| PARAMETER | CONDITIONS | MIN | TYP | MAX | UNITS | NOTES |
|---|------------------------------|-----------|-----------|-----|-------|-------|
| GYROSCOPE SENSITIVITY | | | | | | |
| Full-Scale Range | FS_SEL=0 | ±250 | °/s | | | |
| FS_SEL=1 | ±500 | °/s | | | | |
| FS_SEL=2 | ±1000 | °/s | | | | |
| FS_SEL=3 | ±2000 | °/s | | | | |
| Gyroscope ADC Word Length | 16 | bits | | | | |
| Sensitivity Scale Factor | FS_SEL=0 | 131 | LSB/(°/s) | | | |
| FS_SEL=1 | 65.5 | LSB/(°/s) | | | | |
| FS_SEL=2 | 32.8 | LSB/(°/s) | | | | |
| FS_SEL=3 | 16.4 | LSB/(°/s) | | | | |
| Sensitivity Scale Factor Tolerance | 25°C | -3 | +3 | % | | |
| Sensitivity Scale Factor Variation Over Temperature | ±2 | % | | | | |
| Nonlinearity | Best fit straight line; 25°C | 0.2 | % | | | |
| Cross-Axis Sensitivity | ±2 | % | | | | |
| GYROSCOPE ZERO-RATE OUTPUT (ZRO) | | | | | | |
| Initial ZRO Tolerance | 25°C | ±20 | °/s | | | |
| ZRO Variation Over Temperature | -40°C to +85°C | ±20 | °/s | | | |
| Power-Supply Sensitivity (1-10Hz) | Sine wave, 100mVpp; VDD=2.5V | 0.2 | °/s | | | |
| Power-Supply Sensitivity (10 - 250Hz) | Sine wave, 100mVpp; VDD=2.5V | 0.2 | °/s | | | |
| Power-Supply Sensitivity (250Hz - 100kHz) | Sine wave, 100mVpp; VDD=2.5V | 4 | °/s | | | |
| Linear Acceleration Sensitivity | Static | 0.1 | °/s/g | | | |
| SELF-TEST RESPONSE Relative | Change from factory trim | -14 | 14 | % | 1 | |
| GYROSCOPE NOISE PERFORMANCE | | | | | | |
| Total RMS Noise | DLPFCFG=2 (100Hz) | 0.05 | °/s-rms | | | |
| Low-frequency RMS noise | Bandwidth 1Hz to 10Hz | 0.033 | °/s-rms | | | |
| Rate Noise Spectral Density | At 10Hz | 0.005 | °/s/√ Hz | | | |
| GYROSCOPE MECHANICAL FREQUENCIES | | | | | | |
| X-Axis | 30 | 33 | 36 | kHz | | |
| Y-Axis | 27 | 30 | 33 | kHz | | |
| Z-Axis | 24 | 27 | 30 | kHz | | |
| LOW PASS FILTER RESPONSE | | | | | | |
| Programmable Range | 5 | 256 | Hz | | | |
| ZRO Settling (from power-on) | to ±1°/s of Final | 30 | ms | | | |

Accelerometer Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, TA = 25°C

| PARAMETER | CONDITIONS | MIN | TYP | MAX | UNITS | NOTES |
|--|-----------------------------------|--------|------------------|-----|-------|-------|
| ACCELEROMETER SENSITIVITY | | | | | | |
| Full-Scale Range | AFS_SEL=0 | ±2 | <i>g</i> | | | |
| | AFS_SEL=1 | ±4 | <i>g</i> | | | |
| | AFS_SEL=2 | ±8 | <i>g</i> | | | |
| | AFS_SEL=3 | ±16 | <i>g</i> | | | |
| ADC Word Length | Output in two's complement format | 16 | bits | | | |
| Sensitivity Scale Factor | AFS_SEL=0 | 16,384 | LSB/ <i>g</i> | | | |
| | AFS_SEL=1 | 8,192 | LSB/ <i>g</i> | | | |
| | AFS_SEL=2 | 4,096 | LSB/ <i>g</i> | | | |
| | AFS_SEL=3 | 2,048 | LSB/ <i>g</i> | | | |
| Initial Calibration Tolerance | | ±3 | % | | | |
| Sensitivity Change vs. Temperature | AFS_SEL=0, -40°C to +85°C | ±0.02 | %/°C | | | |
| Nonlinearity | Best Fit Straight Line | 0.5 | % | | | |
| Cross-Axis Sensitivity | | ±2 | % | | | |
| ZERO-G OUTPUT | | | | | | |
| Initial Calibration Tolerance | X and Y axes | ±50 | <i>mg</i> | 1 | | |
| Z axis | | ±80 | <i>mg</i> | | | |
| Zero-G Level Change vs. Temperature | X and Y axes, 0°C to +70°C | ±35 | | | | |
| | Z axis, 0°C to +70°C | ±60 | <i>mg</i> | | | |
| SELF TEST RESPONSE | | | | | | |
| Relative | Change from factory trim | -14 | 14 | % | 2 | |
| NOISE PERFORMANCE | | | | | | |
| Power Spectral Density | @10Hz, AFS_SEL=0 & ODR=1kHz | 400 | μ <i>g</i> /√ Hz | | | |
| LOW PASS FILTER RESPONSE | | | | | | |
| Programmable Range | | 5 | 260 | Hz | | |
| OUTPUT DATA RATE | | | | | | |
| Programmable Range | | 4 | 1,000 | Hz | | |
| INTELLIGENCE FUNCTION INCREMENT | | | | | | |
| | | 32 | <i>mg</i> /LSB | | | |

Electrical and Other Common Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, TA = 25°C

| PARAMETER | CONDITIONS | MIN | TYP | MAX | Units | Notes |
|---|--|------|--------|-----|-------|-------|
| TEMPERATURE SENSOR | | | | | | |
| Range | -40 to +85 | °C | | | | |
| Sensitivity | Untrimmed | 340 | LSB/°C | | | |
| Temperature Offset | 35°C | -521 | LSB | | | |
| Linearity | Best fit straight line (-40°C to +85°C) | ±1 | °C | | | |
| VDD POWER SUPPLY | | | | | | |
| Operating Voltages | 2.375 | 3.46 | V | | | |
| Normal Operating Current | Gyroscope + Accelerometer + DMP | 3.9 | mA | | | |
| Gyroscope + Accelerometer (DMP disabled) | 3.8 | mA | | | | |
| Gyroscope + DMP (Accelerometer disabled) | 3.7 | mA | | | | |
| Gyroscope only (DMP & Accelerometer disabled) | 3.6 | mA | | | | |
| Accelerometer only (DMP & Gyroscope disabled) | 500 | µA | | | | |
| Accelerometer Low Power Mode Current | 1.25 Hz update rate | 10 | µA | | | |
| 5 Hz update rate | 20 | µA | | | | |
| 20 Hz update rate | 70 | µA | | | | |
| 40 Hz update rate | 140 | µA | | | | |
| Full-Chip Idle Mode Supply Current | 5 | µA | | | | |
| Power Supply Ramp Rate | Monotonic ramp. Ramp rate is 10% to 90% of the final value | 100 | ms | | | |
| VLOGIC REFERENCE VOLTAGE | | | | | | |
| Voltage Range | VLOGIC must be ≤VDD at all times | 1.71 | VDD | V | | |
| Power Supply Ramp Rate | Monotonic ramp. Ramp rate is 10% to 90% of the final value | 3 | ms | | | |
| Normal Operating Current | 100 | µA | | | | |
| TEMPERATURE RANGE | | | | | | |
| Specified Temperature Range | Performance parameters are not applicable beyond Specified Temperature Range | -40 | +85 | °C | | |

Electrical Specifications, Continued

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, TA = 25°C

| PARAMETER | CONDITIONS | MIN | TYP | MAX | Units | Notes |
|---|---|----------|-----|-----|-------|-------|
| SERIAL INTERFACE | | | | | | |
| SPI Operating Frequency, All Registers Read/Write | MPU-6000 only, Low Speed Characterization | 100 ±10% | kHz | | | |
| MPU-6000 only, High Speed Characterization | 1 ±10% | MHz | | | | |
| SPI Operating Frequency, Sensor and Interrupt Registers Read Only | MPU-6000 only | 20 ±10% | MHz | | | |
| I ² C Operating Frequency | All registers, Fast-mode | 400 | kHz | | | |
| All registers, Standard-mode | 100 | kHz | | | | |
| I²C ADDRESS | | | | | | |
| AD0 = 0 | | 1101000 | | | | |
| AD0 = 1 | | 1101001 | | | | |
| DIGITAL INPUTS (SDI/SDA, AD0, SCLK/SCL, FSYNC, /CS, CLKIN) | | | | | | |
| V _{IH} , High Level Input Voltage | MPU-6000 | 0.7*VDD | V | | | |
| MPU-6050 | 0.7*VLOGIC | V | | | | |
| V _{IL} , Low Level Input Voltage | MPU-6000 | 0.3*VDD | V | | | |
| MPU-6050 | 0.3*VLOGIC | V | | | | |
| C _i , Input Capacitance | < 5 | pF | | | | |
| DIGITAL OUTPUT (SDO, INT) | | | | | | |
| V _{OH} , High Level Output Voltage | R _{LOAD} =1MΩ; MPU-6000 | 0.9*VDD | V | | | |
| R _{LOAD} =1MΩ; MPU-6050 | 0.9*VLOGIC | V | | | | |
| V _{OL1} , LOW-Level Output Voltage | R _{LOAD} =1MΩ; MPU-6000 | 0.1*VDD | V | | | |
| R _{LOAD} =1MΩ; MPU-6050 | 0.1*VLOGIC | V | | | | |
| V _{OLINT1} , INT Low-Level Output Voltage | OPEN=1, 0.3mA sink Current | 0.1 | V | | | |
| Output Leakage Current | OPEN=1 | 100 | nA | | | |
| t _{INT} , INT Pulse Width | LATCH_INT_EN=0 | 50 | μs | | | |

Electrical Specifications, Continued

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5%
or

VDD, TA = 25°C

| Parameters | Conditions | Typical | Units | Notes |
|--|--------------------------------------|-----------------------------|-------|-------|
| Primary I²C I/O (SCL, SDA) | | | | |
| V _{IL} , LOW-Level Input Voltage | MPU-6000 | -0.5 to 0.3*VDD | V | |
| V _{IH} , HIGH-Level Input Voltage | MPU-6000 | 0.7*VDD to VDD + 0.5V | V | |
| V _{hys} , Hysteresis | MPU-6000 | 0.1*VDD | V | |
| V _{IL} , LOW Level Input Voltage | MPU-6050 | -0.5V to 0.3*VLOGIC | V | |
| V _{IH} , HIGH-Level Input Voltage | MPU-6050 | 0.7*VLOGIC to VLOGIC + 0.5V | V | |
| V _{hys} , Hysteresis | MPU-6050 | 0.1*VLOGIC | V | |
| V _{OL1} , LOW-Level Output Voltage | 3mA sink current | 0 to 0.4 | V | |
| I _{OL} , LOW-Level Output Current | V _{OL} = 0.4V | 3 | mA | |
| | V _{OL} = 0.6V | 5 | mA | |
| Output Leakage Current | 100 | nA | | |
| t _{of} , Output Fall Time from V _{IHmax} to V _{ILmax} | C _b bus capacitance in pF | 20+0.1C _b to 250 | ns | |
| C _i , Capacitance for Each I/O pin | < 10 | pF | | |
| Auxiliary I²C I/O (AUX_CL, AUX_DA) | MPU-6050: AUX_VDDIO=0 | | | |
| V _{IL} , LOW-Level Input Voltage | -0.5V to 0.3*VLOGIC | V | | |
| V _{IH} , HIGH-Level Input Voltage | 0.7*VLOGIC to VLOGIC + 0.5V | V | | |
| V _{hys} , Hysteresis | 0.1*VLOGIC | V | | |
| V _{OL1} , LOW-Level Output Voltage | VLOGIC > 2V; 1mA sink current | 0 to 0.4 | V | |
| V _{OL3} , LOW-Level Output Voltage | VLOGIC < 2V; 1mA sink current | 0 to 0.2*VLOGIC | V | |
| I _{OL} , LOW-Level Output Current | V _{OL} = 0.4V | 1 | mA | |
| | V _{OL} = 0.6V | 1 | mA | |
| Output Leakage Current | 100 | nA | | |
| t _{of} , Output Fall Time from V _{IHmax} to V _{ILmax} | C _b bus capacitance in pF | 20+0.1C _b to 250 | ns | |
| C _i , Capacitance for Each I/O pin | < 10 | pF | | |

Electrical Specifications, Continued

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5%
or

VDD, TA = 25°C

| Parameters | Conditions | Min | Typical | Max | Units | Notes |
|--------------------------------------|--|------------|---------|-----|-------|-------|
| INTERNAL CLOCK SOURCE | CLK_SEL=0,1,2,3 | | | | | |
| Gyroscope Sample Rate, Fast | DLPFCFG=0 SAMPLERATEDIV = 0 | 8 | kHz | | | |
| Gyroscope Sample Rate, Slow | DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0 | 1 | kHz | | | |
| Accelerometer Sample Rate | 1 | kHz | | | | |
| Clock Frequency Initial Tolerance | CLK_SEL=0, 25°C | -5 | +5 | % | | |
| CLK_SEL=1,2,3; 25°C | -1 | +1 | % | | | |
| Frequency Variation over Temperature | CLK_SEL=0 | -15 to +10 | % | | | |
| CLK_SEL=1,2,3 | ±1 | % | | | | |
| PLL Settling Time | CLK_SEL=1,2,3 | 1 | 10 | ms | | |
| EXTERNAL 32.768kHz CLOCK | CLK_SEL=4 | | | | | |
| External Clock Frequency | 32.768 | kHz | | | | |
| External Clock Allowable Jitter | Cycle-to-cycle rms | 1 to 2 | µs | | | |
| Gyroscope Sample Rate, Fast | DLPFCFG=0 SAMPLERATEDIV = 0 | 8.192 | kHz | | | |
| Gyroscope Sample Rate, Slow | DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0 | 1.024 | kHz | | | |
| Accelerometer Sample Rate | 1.024 | kHz | | | | |
| PLL Settling Time | 1 | 10 | ms | | | |
| EXTERNAL 19.2MHz CLOCK | CLK_SEL=5 | | | | | |
| External Clock Frequency | 19.2 | MHz | | | | |
| Gyroscope Sample Rate | Full programmable range | 3.9 | 8000 | Hz | | |
| Gyroscope Sample Rate, Fast Mode | DLPFCFG=0 SAMPLERATEDIV = 0 | 8 | kHz | | | |
| Gyroscope Sample Rate, Slow Mode | DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0 | 1 | kHz | | | |
| Accelerometer Sample Rate | 1 | kHz | | | | |
| PLL Settling Time | 1 | 10 | ms | | | |

Applications Information

Pin Out and Signal Description

| Pin Number | MPU-6000 | MPU-6050 | Pin Name | Pin Description |
|----------------------------|----------|------------|--|--|
| 1 | Y | Y | CLKIN | Optional external reference clock input. Connect to GND if unused. |
| 6 | Y | Y | AUX_DA | I ² C master serial data, for connecting to external sensors |
| 7 | Y | Y | AUX_CL | I ² C Master serial clock, for connecting to external sensors |
| 8 | Y | /CS | SPI chip select (0=SPI mode) | |
| 8 | Y | VLOGIC | Digital I/O supply voltage | |
| 9 | Y | AD0 / SDO | I ² C Slave Address LSB (AD0); SPI serial data output (SDO) | |
| 9 | Y | AD0 | I ² C Slave Address LSB (AD0) | |
| 10 | Y | Y | REGOUT | Regulator filter capacitor connection |
| 11 | Y | Y | FSYNC | Frame synchronization digital input. Connect to GND if unused. |
| 12 | Y | Y | INT | Interrupt digital output (totem pole or open-drain) |
| 13 | Y | Y | VDD | Power supply voltage and Digital I/O supply voltage |
| 18 | Y | Y | GND | Power supply ground |
| 19, 21 | Y | Y | RESV | Reserved. Do not connect. |
| 20 | Y | Y | CPOUT | Charge pump capacitor connection |
| 22 | Y | Y | RESV | Reserved. Do not connect. |
| 23 | Y | SCL / SCLK | I ² C serial clock (SCL); SPI serial clock (SCLK) | |
| 23 | Y | SCL | I ² C serial clock (SCL) | |
| 24 | Y | SDA / SDI | I ² C serial data (SDA); SPI serial data input (SDI) | |
| 24 | Y | SDA | I ² C serial data (SDA) | |
| 2, 3, 4, 5, 14, 15, 16, 17 | Y | Y | NC | Not internally connected. May be used for PCB trace routing. |