

AUTOMAÇÃO DE UMA MÁQUINA DE COSTURA PARA REALIZAR BORDADOS

José Rodolfo Mantovani (FHO - Uniararas) j-rodolfomantovani@alunos.fho.edu.br
Mauricio Acconcia Dias (FHO - Uniararas) macdias@fho.edu.br

Resumo

A automação está presente em nosso cotidiano em muitos casos é muito alto o custo para seu desenvolvimento, mas para quase todo projeto existe uma solução de baixo custo que pode ser aplicado com o uso da tecnologia Arduino que se trata de uma plataforma *open source* capaz de proporcionar um desenvolvimento rápido e com inúmeras soluções. No mercado existem máquinas doméstica de bordados com custos de aquisição elevados, tendo isso em vista, o trabalho tem como objetivo automatizar uma máquina de costura convencional para realizar bordados computadorizados, ela receberá comandos de um dispositivo *mobile* na plataforma Android que é o interpretador dos arquivos de bordados e envia para o Arduino na conexão *Bluetooth*, onde atuará como controlador do projeto, assim, consequentemente, moverá a mesa pela ação dos motores de passos, confeccionando o desenho programado.

Palavras-Chave: arduino, cnc, automação, máquina de bordado

1. Introdução

Conforme definido por Rosário (2005), onde se encontra uma economia globalizada, a capacidade de inovar, buscar nova ideias e absorver opiniões vale mais do que a gestão de ativos físicos. Portanto, a capacidade de descobrir novas necessidades e encontrar soluções para elas induz a indústria a se atualizar e modernizar sua maneira de agir em seus produtos, desta forma umas das opções de alcançar esse caminho é a automação.

Um projeto de automação pode gerar um alto custo em seu desenvolvimento, ficando inviável sua realização. No entanto, com a popularização do Arduino, uma plataforma *open source* sendo *hardware* de baixo custo onde se pode fazer muitos projetos, como uma automação residencial, irrigação automatizada de plantas, entre outros projetos. No mercado, existem máquinas de bordados domésticas com altos valores e neste projeto foi avaliado a possibilidade da utilização do Arduino como uma alternativa para redução dos valores investidos.

O objetivo deste trabalho é o desenvolvimento e automação de uma máquina de costura para confecção de bordados computadorizados no âmbito de ser uma solução de baixo custo. Para a parte de controle será utilizado o Arduino junto com a *CNC Shield*, que receberá um sinal de

um dispositivo *mobile* na plataforma Android, assim sendo interpretada pelos comandos recebidos, consequentemente acionado os motores de passo, assim confeccionando o desenho de bordado. Sendo estes comandos CNC (Comando Numérico Computadorizado) gerado pelo aplicativo por meio da leitura do arquivo de bordado, identificando as coordenadas do desenho e converte para *G-code* para posteriormente ser enviada.

2. Revisão bibliográfica

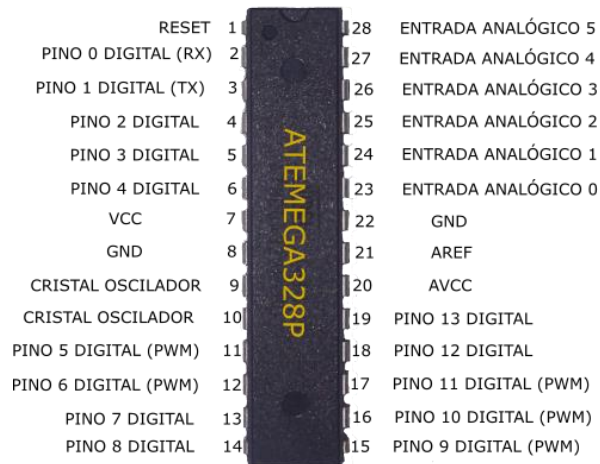
A automação está presente no nosso dia-a-dia, seja na indústria ou em residências, tornando-se algo muito importante para sociedade. Conforme observado por Stevam JR e Silva (2015), para onde se pode olhar quase sempre há algo sendo controlado, seja no controle de luminosidade, no controle de umidade, na detecção de elementos em certas posição, no controle em níveis de temperaturas, no controle de movimentos de motores, entre outros sistemas, pois estes requerem mecanismos de controle extremamente confiáveis e na maioria dos casos com custos elevados, mas, por outro lado, em quase todos os exemplos mencionados poderia ser aplicado uma solução de baixo custo, aplicando microcontroladores e uma alternativa para isso é a plataforma Arduino.

2.1. Arduino

Segundo McRoberts (2011), o Arduino pode ser definido como um pequeno computador onde é possível programar para processar dados de entradas e saídas de dispositivos e componentes conectado a ele. A placa de Arduino pode ser classificado como computação embarcada, sendo um sistema capaz de interagir com o meio em sua volta através de hardware e software.

O Arduino possui vários modelos, mas neste artigo será abordado o Arduino Uno, pois o mesmo foi utilizado no projeto. Trata-se de uma placa que utiliza um microcontrolador ATMEGA 328p, possuindo 14 pinos de entrada e saídas digital, sendo 6 pinos que podem ser configurados para fornecer uma função PWM (modulação por largura de pulso) em suas saídas, possuindo também 6 entradas analógicas e um cristal oscilador de 16 MHz. Na figura 1, abaixo, podemos observar os pinos do microcontrolador ATMEGA328p.

Figura 1- ATMEGA328p



Fonte: Próprio autor

Conforme definido por Evans, Noble e Hochenbaum(2013) as *Shields* (escudos) para Arduino são placas eletrônicas que se encaixam no Arduino adicionando novas funções. A maior parte delas necessita importar uma biblioteca no Arduino para sua utilização e também algumas *shields* têm seus próprios conectores onde pode ser acoplado outras *shields*. Segundo Stevam JR e Silva (2015), por existir uma padronização em seu formato e pinagens, a *shield* pode ser facilmente acoplado no Arduino e também e outras *shields*, formando um empilhamento de *shields*, sendo assim reduzindo a necessidade de possuir conectores para ligação entre os módulos e componentes, agilizando de maneira prática o desenvolvimento de projetos envolvendo Arduino em sua etapa preliminar.

Podemos citar alguns exemplos de Shields como Ethernet *Shield*, permitindo o Arduino se conectar com a internet, *shield* 3g com gps permitindo que o Arduino mostre sua localização no globo terrestre, Motor *Shield* destinado na utilização de motores de corrente contínua e muitos outros modelos existentes, sendo que a partir da necessidade pode se criar sua própria *shield* tendo em vista que foi utilizado em projeto A e quer utilizar em um projeto B, basta projetar em uma placa e soldar os componentes e utilizar como *shield* no Arduino.

2.2. Controle Numérico Computadorizado (CNC)

De acordo com Suh et al (2008), os sinais enviados de um controlador convertendo em movimento mecânico são de extrema importância na concepção de um bom trabalho, assim sendo indispensável para garantir controle e precisão dos movimentos realizados por máquina

CNC. Uns elementos que se destacam são os motores de passo encarregado de converter sinais elétricos em movimento, fusos e polia dando movimentação a mesa ou eixo móvel da máquina.

Segundo Silvia (2002), para as máquinas CNC existe um padrão, com regras definidas, sobre programação que são *G-code*, dando a liberdade para criação de comandos, sendo assim possui vários comandos que aparentemente fazem a mesma coisa, porém possuem mais recurso que outros.

2.3. Motor de passo

A programação *G-code* é uma linguagem bem simplificada e as máquinas CNC interpretam esses códigos de forma sequencial, linha por linha. Assim, envia comandos para os motores de passos e consequentemente tem movimentos precisos.

Motores de passo são motores específicos com características de se mover em uma sequência de passos discretos, sendo uma escolha ideal para projetos que requerem um controle e precisão em seus movimentos. (EVANS, NOBLE E HOCHENBAUM, 2013).

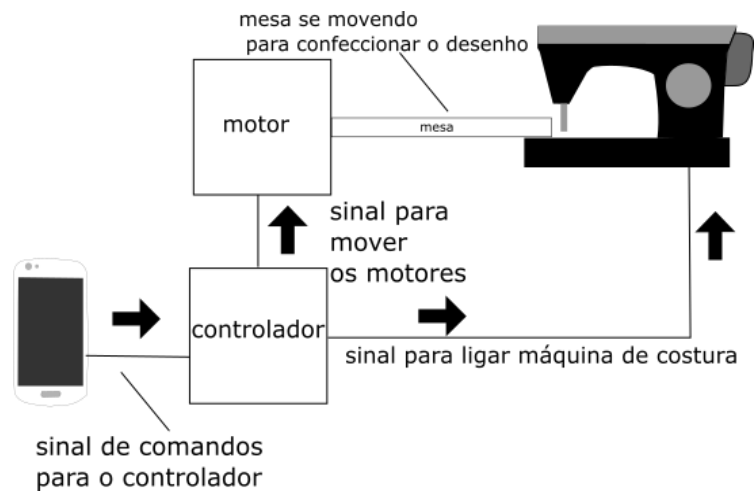
3. Matérias e métodos

Este projeto consiste-se na utilização dos seguintes componentes para seu desenvolvimento:

- **Hardware:** Arduino Uno R3, CNC shield V3, drive A4988, módulo relé, motor de passo Nema 17 e módulo *Bluetooth*.
- **Firmware:** grbl.
- **Software:** *Universal Gcode*.
- **Interface para usuário:** aplicativo *mobile* da plataforma Android.
- **Estrutura da mesa:** perfil de alumino, base de madeira e uma chapa de ferro.
- **Máquina:** Máquina de costura Singer *zig zag super*, um *dimmer*, e um bastidor.

Como pode ser observado na figura 2, encontra-se um croqui do sistema, onde um dispositivo na plataforma Android envia um comando *g-code* para o controlador que é o Arduino através de uma comunicação *Bluetooth*, assim com auxílio de CNC *shield* através dos drives manda sinal elétrico para motor que é convertido e movimento mecânico.

Figura 2- Croqui do sistema

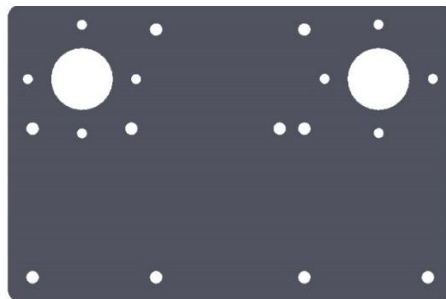


Fonte: Próprio autor

3.1. Estrutura da mesa

Na estrutura da mesa constitui-se com uma chapa de ferro galvanizado que pode ser observado na figura 3, possuindo dimensões 18x12cm onde nela fixou-se os componentes como motor e as roldanas.

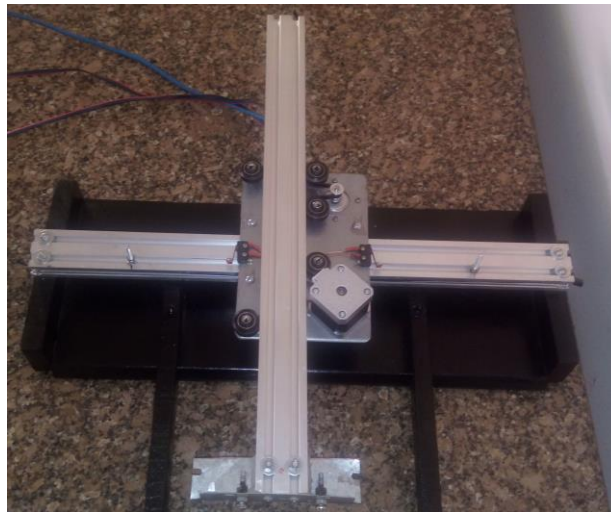
Figura 3 – Desenho da chapa de ferro galvanizado da mesa



Fonte: Adaptado OpenBuilds (2014)

Nas roldanas encachou-se o perfil de alumínio, sendo que o eixo X, ou seja, eixo horizontal fixou-se na base de madeira, e o eixo Y move livre na mesa, sendo que correia foi fixado na no perfil do alumínio, onde podemos ver na figura 4 a montagem da mesa por completo.

Figura 4 – Mesa montada



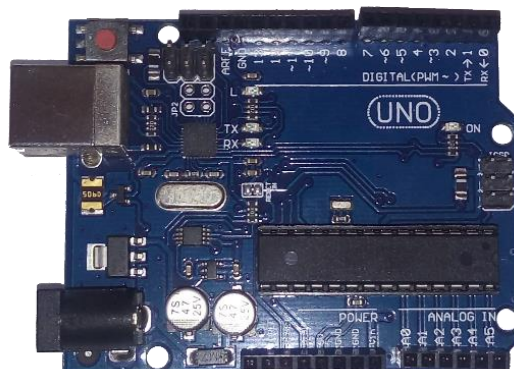
Fonte: Próprio autor

3.2. Hardware

3.2.1. Arduino

Utilizou-se o Arduino Uno R3 sendo uma plataforma open source, que pode ser observado na figura 5, onde o Arduino atua-se como controlador do projeto, mas o que promove o seu funcionamento é o microcontrolador ATmega 328p, possui 14 pinos de entrada e saídas digital, do qual 6 podem ser usados como saídas PWM, 6 entradas alógicas, um cristal oscilador de quartzo de 16MHz, uma conexão USB, um conector de energia e um botão *reset*.

Figura 5 – Arduino Uno R3



Fonte: Próprio autor

Em resumo pode ser observado as principais características do Arduino na tabela 1.

Tabela 1 – Principais características do Arduino UNO

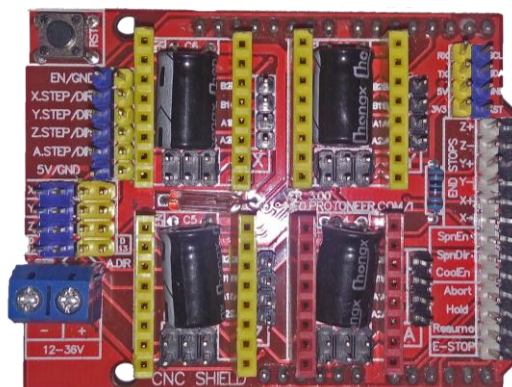
Microcontrolador	ATmega328P
Tensão operacional	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limite)	6-20 V
Pinos Digital I/O	14 (dos quais 6 fornecem saída PWM)
Pinos de entrada analógica	6
Corrente DC por pino de I/O	20 mA
Corrente DC para pino 3.3V	50 mA
Memória flash	32 KB (ATmega328P) dos quais 0,5 KB usados pelo bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidade do clock	16 megahertz

Fonte: Adaptado de Arduino (2018)

Para projeto utilizou-se a CNC *shield* v3 como *shield* (escudo) trazendo funcionalidade para o Arduino de controlar motores de passo, assim a CNC *shield* V3 facilita-se a conexão dos drives do motor com Arduino, também necessita-se de uma biblioteca seja carregada para seu completo funcionamento, pois ela suporta 4 slots par conexão de drivers A49988, ou seja, ela suporta até 4 motores de passos sendo que cada motor utiliza duas portas de entrada e saídas. Pode suportar uma tensão de entrada para motores de 12V a 36V CC (Corrente Continua), contando também com 2 conectores de fim de curso para cada eixo.

Na figura 6 exibe a CNC *shield* com seus respectivos pinos.

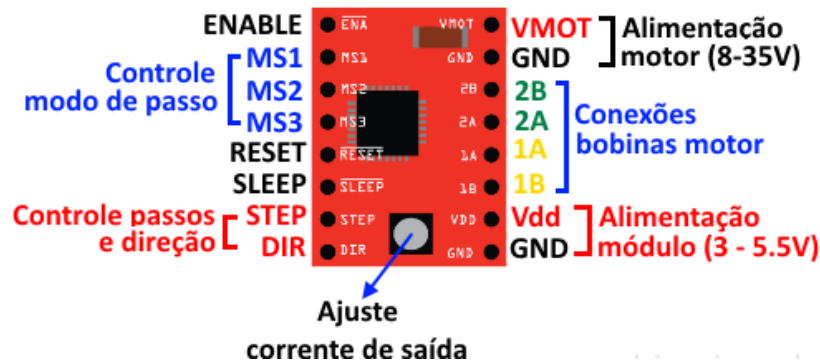
Figura 6 – CNC shield V3



Fonte: Próprio autor

O drive A4988 possui três pinos para combinações que pode fazer com jumpers no slot da CNC *shield* para configurar micropassos do motor que podemos ver na figura 7 os pinos MS1, MS2 e MS3 do drive A4988, e associando os em combinação este pinos, podemos ver o resultado na tabela 2 que são modos *full-step*, *half-step*, *quater(1/4)*, *eight step(1/8)* e *sixteenth setp(1/16)*.

Figura 7 – Pinos drive A4988



Fonte: Arduino e Cia (2015)

Tabela 2 – Combinações dos pinos na CNC *shield*

MS1	MS2	MS3	Configuração micropassos
<i>Low</i>	<i>Low</i>	<i>Low</i>	<i>Full step</i>
<i>High</i>	<i>Low</i>	<i>Low</i>	<i>Half step</i>
<i>Low</i>	<i>High</i>	<i>Low</i>	<i>Quarter step(1/4)</i>
<i>High</i>	<i>High</i>	<i>Low</i>	<i>Eighth step (1/8)</i>
<i>High</i>	<i>High</i>	<i>High</i>	<i>Sixteth sep(1/16)</i>

Fonte: Adaptado de Arduino e Cia (2015)

Na figura 7 acima onde apresenta as pinagens do drive A4988 tem um *trimpot* (um pequeno potenciômetro) onde se faz o ajuste de corrente de saída para motor, para realizar o ajuste é necessário medir a tensão em cima do *trimpot*, e para achar o valor ser ajustado é preciso fazer um pequeno cálculo.

$$I_{Max} = \frac{V_{REF}}{8 * R_{CS}} \quad (1)$$

Onde I_{max} é corrente máxima do motor; R_{CS} é resistor de *sense* do drive A4988; V_{REF} o valor da tensão que precisa ser ajustado no *trimpot*.

É recomendada pela própria *datasheet* não utilizar a corrente máxima do motor e sim utilizar 70% dela, pois senão obrigaria o motor a trabalhar no seu limite.

3.2.2. Motor de passo

Utilizou-se motor de passo Nema 17, modelo 17HS4401 possuindo uma corrente nominal de 1,7A tendo como ângulo de passo de 1.8° que pode ser observado na figura 8.

Figura 8 – Motor de Passo Nema 17

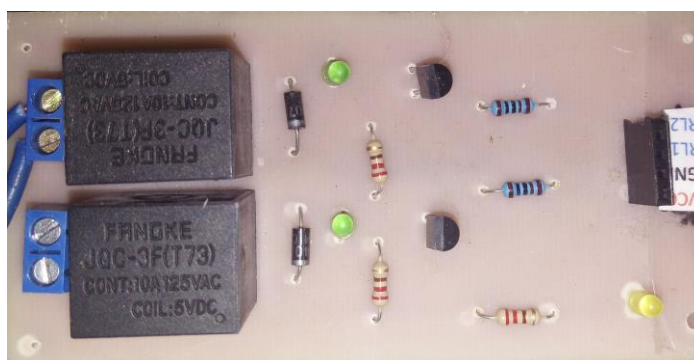


Fonte: Próprio Autor

3.2.3. Módulo relé

Utilizou-se um módulo rele que basicamente tem função de acionar a máquina de costura, recebendo nos pinos a alimentação Vcc e GND e o sinal vindo do Arduino no RL2 para ligar e desligar máquina que pode ser observado com mais detalhe na figura 9, os componentes utilizados no modulo relé foi um relé 5 v, 2 led, 2 resistores de 220Ω , um resistor de $1k\Omega$, um transistor BC5488 e um diodo 1n4007 para um módulo com 1 relé no caso abaixo utilizou-se um módulo de com dois relé, pois foi-se reaproveitados de outros projeto, mas não há necessidade de utiliza um módulo com dois relés.

Figura 9 – Módulo relé

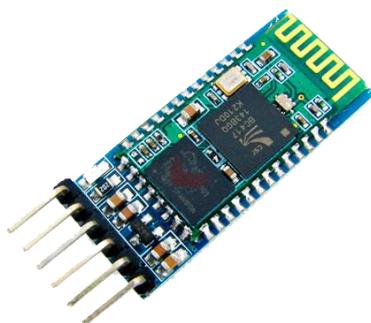


Fonte: Próprio autor

3.2.4. Módulo *Bluetooth*

A comunicação constitui-se na utilização do módulo *Bluetooth* HC-05 fazendo o meio campo entre o Arduino e dispositivo *mobile* Android. É preciso alterar a taxa de transmissão do módulo para 115200, pois o mesmo que está sendo utilizado no Arduino, por padrão vem configurado 9600. Entre os pinos RX do módulo *Bluetooth* e TX localizado CNC *shield* que o mesmo do Arduino precisa realizar - se uma divisão de tensão, devido a tensão no pino RX do módulo não pode ultrapassar os 3,3V podendo danificar, sendo assim montou-se uma pequena PCI (Placa de Circuito Impresso) com o módulo *Bluetooth* e dois resistores de um de 1k Ω e dois resistores de 1k Ω em série formando 2 K Ω , tendo uma queda de tensão 3,3V, que pode ser observado o módulo Bluetooth na figura 10.

Figura 10 – Módulo *Bluetooth*

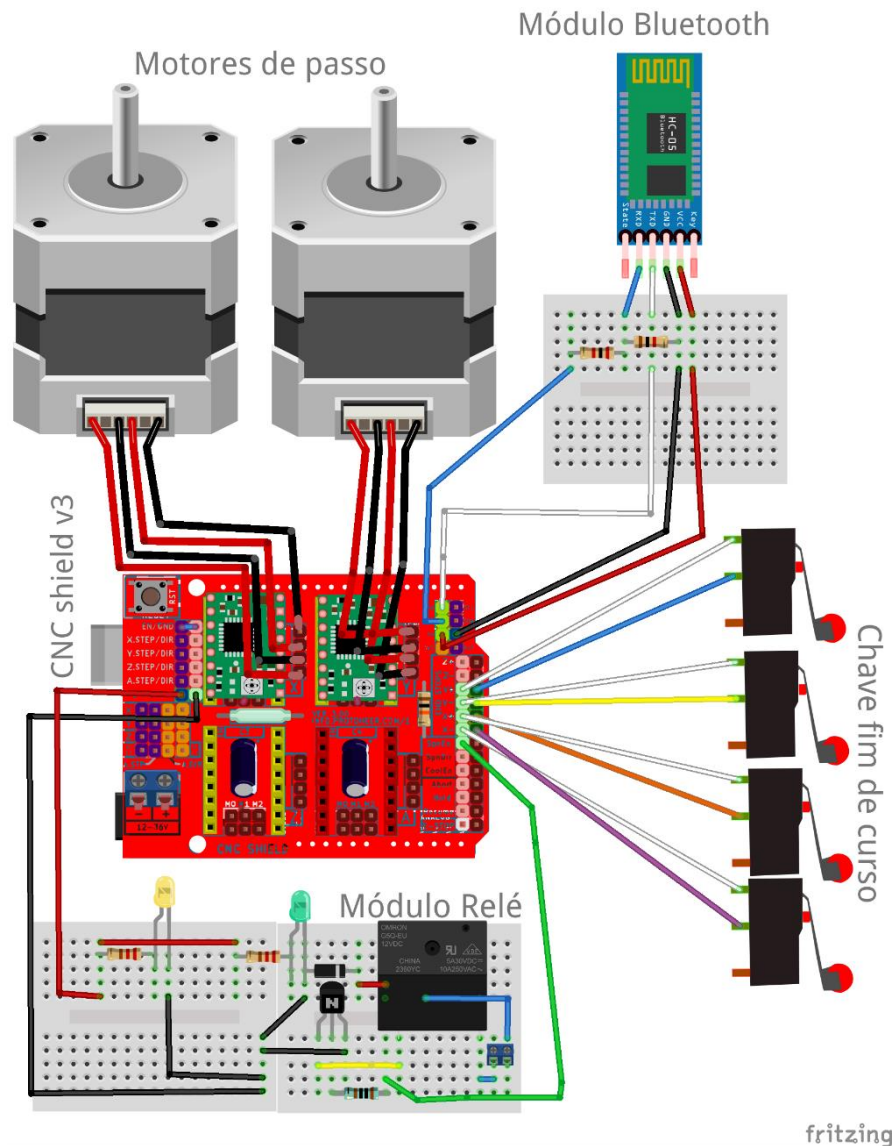


Fonte: Próprio Autor

3.2.5. Esquema elétrico

Na figura 11 abaixo podemos ver esquema elétrico do projeto, onde temos o Arduino Uno R3 devidamente acoplado com a CNC *shield* v3 e também inseridos em seus slots os drives que atribui uma nova função que é aquela na qual controlar motor de passos, com apenas dois dos seus slots ocupados sendo X e Y, pois neste projeto não utilizaremos o eixo Z e também o eixo A destinado para replicar os movimentos de uns dos eixos. Para ligação dos motores na CNC *shield* não precisou identificar as bobinas A e B pois os mesmos vieram com os cabos configurados, apenas bastou-se conecta-los na CNC *shield*, mas com uma ressalva com o conector do eixo Y ficou - se invertido em relação ao eixo X, para que a direção do eixo Y tivesse seu sentido trocado, assim evitando de configura via *software*.

Figura 11 – Esquema elétrico



Fonte: Próprio autor

Na figura 11 acima podemos ver a ligação do módulo *Bluetooth* e também do módulo relé onde os pinos RX, TX, VCC e GND do módulo *Bluetooth* deve se conectar com pinos da TX, RX, VCC E GND respectivamente com uma divisão tensão resultando 3,3 V no pino RX e o módulo relé recebe um sinal vindo do Arduino através do pino SpnEn da CNC *shield* para acionamento da máquina de costura.

3.3. *Firmware Grbl*

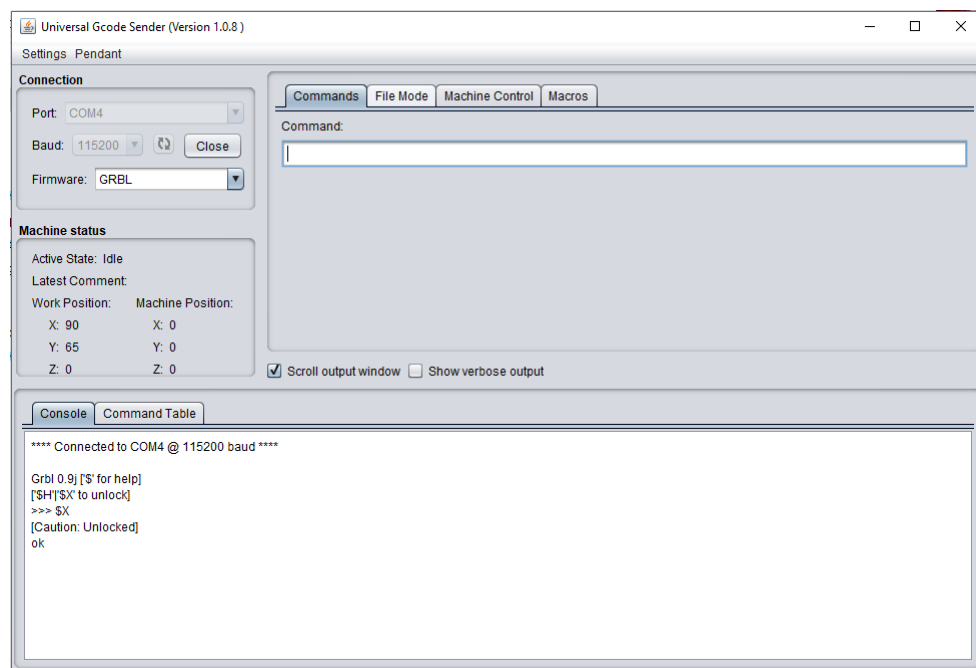
O Grbl é um *firmware* interpretador de G-code desenvolvido para Arduino na linguagem C, onde é carregado dentro do microcontrolador do Arduino destinado a controlar os motores de passo. Utilizou - se a versão 0.9j do Grbl tendo uma taxa de comunicação serial de 115200,

podemos acessar ele via cabo USB para configurar a velocidade e calibrar o passo dos motores utiliza-se um software que é o Universal Gcode.

3.4. Software Universal Gcode

Para fazer a configuração do Grbl utilizou-se o *Universal Gcode* que pode ser observado na figura 12, que se conecta ao computador via cabo USB o Arduino, mas antes precisa se certificar está com módulo *Bluetooth* desconectado, senão não há comunicação, e nele também é possível enviar comandos para máquina.

Figura 12 – Universal Gcode



Fonte: Próprio autor

Uma dessa configurações necessita-se de um cálculo que é passo por milímetro, utiliza-se a seguinte fórmula destinada para sistema de tração por correia e polia:

$$\frac{\text{Step}}{\text{mm}} = \frac{\text{configuração de micropassos} * \text{Passos por volta do motor}}{\text{distância do dente da correia} * \text{número dentes da polia}} \quad (2)$$

3.5. Interface do usuário

A interface do usuário é um aplicativo da plataforma Android, onde ele é um visualizador de bordados disponível no GitHub com nome *Mobile Embroidery Viewer*, tem função de abrir um arquivo de bordado e exibir o desenho na tela do aparelho celular. Utilizando-se IDE Android Studio com linguagem Java e XML, neste aplicativo adaptando para pegar as coordenadas lidos do arquivo de bordado e gerando comando *G-code* para ser enviado ao Arduino.

Figura 13 – Imagem da tela do aplicativo sem modificação



Fonte: Próprio autor

3.6. Máquina de costura

Utilizou-se uma máquina de costura Singer *zig zag super*, possuindo funções de costura doméstica, necessitando de controle velocidade utilizou - se um *dimmer* mantendo a velocidade da máquina constante, também se utilizou um bastidor de bordado com área de 18x13 cm, onde prende-se o tecido para ser bordado.

Figura 14 – Máquina de costura



Fonte: Próprio autor

4. Resultados

Para configuração micropassos optou-se ¼ de passo, colocando o jumper no pino do meio do slot da CNC *shield* nos eixos X e Y, pois o mesmo apresentou-se uma boa redução e houve uma pequena perda torque do motor, mas ainda possuindo bom desempenho e apresentando uma movimentação precisa da mesa.

Para iniciar o cálculo de ajuste da corrente de saída primeiros levantou -se os dados necessários do motor de passo e do drive sendo corrente 1,7A do motor e resistor de *sense* com valor 0,1 ohms do drive.

Desta forma a formula ficou assim:

$$V_{REF} = I_{Max} * 8 * R_{CS} \quad (3)$$

Aplicando-se a formula o com 70% da corrente do motor que se aproximou-se 1,2A, obteve-se um V_{REF} de 960mV, com auxílio de multímetro e uma pequena chave configurou-se girando *trimpot* do drive A4998.

Para calcular passo por milímetro como definido na configuração de micropassos de ¼, utiliza-se 4 no cálculo, para passos por volta do motor é definido pela divisão de 360° pelo grau do motor que de 1,8° resultando num total de 200 passos, distância de cada dente da correia que é 2 mm, e número de dentes da polia é 16.

Então a aplicando-se formula de sistema de tração por correia e polia resultando - se no valor a ser configurando:

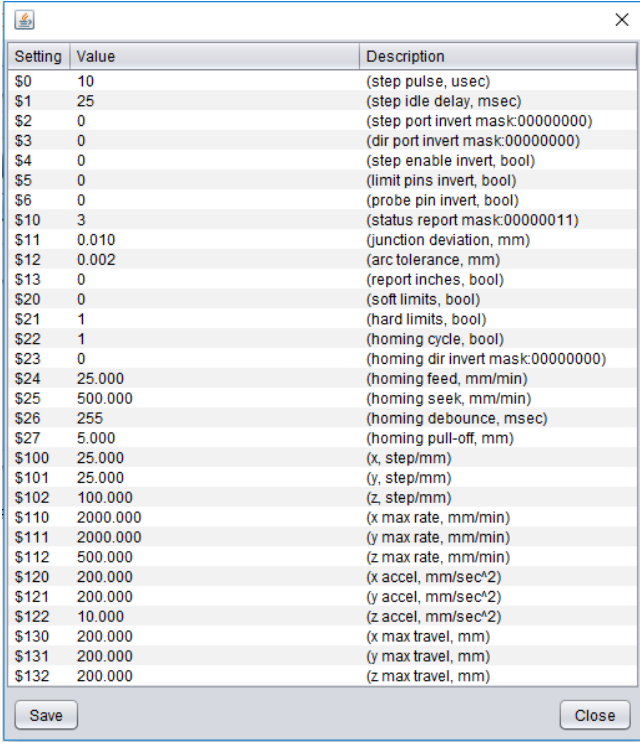
$$\frac{Step}{mm} = \frac{4 * 200}{2 * 16} = 25 \text{ step/mm}$$

Acessando-se as configurações no Grbl onde é exibido uma tela com os parâmetros pode ser observado na figura 15, alterou-se os seguintes parâmetros:

- \$21 passando valor de 0 para 1 assim habilitando uso de chave de fim curso.
- \$22 passando de valor 0 para 1 assim habilitando que máquina use o fim de curso para determinar qual posição vai iniciar quando se perde a referência do ponto 0 da máquina, o ponto 0 definido por comando quando se aperta o botão *homing*.

- \$100 e \$101 correspondentes ao eixo X e Y respectivamente, onde se configura o passo por milímetros que significa quantos passo o motor tem de fazer para andar um milímetro, atribuiu-se um valor de 25 passos/mm, definido pelo cálculo acima.
- \$110 e \$111 são taxa máxima para que cada eixo possa se mover que caso são o eixo X e Y que atribui - se um valor de 2000 mm/min.
- \$120 e \$121 são a taxa de aceleração dos eixos X e Y respetivamente, dado por aceleração em milímetro por segundo ao quadrado, que vai gerar movimentos rápidos e lentos dependendo do valor passado para eixo. Atribuiu-se os valores para o eixo X e Y de uma aceleração de 200 mm/s².
- O restante dos parâmetros permaneceu com valores padrão.

Figura 15 – Parâmetros de configuração do Grbl



Setting	Value	Description
\$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	0.010	(junction deviation, mm)
\$12	0.002	(arc tolerance, mm)
\$13	0	(report inches, bool)
\$20	0	(soft limits, bool)
\$21	1	(hard limits, bool)
\$22	1	(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	255	(homing debounce, msec)
\$27	5.000	(homing pull-off, mm)
\$100	25.000	(x, step/mm)
\$101	25.000	(y, step/mm)
\$102	100.000	(z, step/mm)
\$110	2000.000	(x max rate, mm/min)
\$111	2000.000	(y max rate, mm/min)
\$112	500.000	(z max rate, mm/min)
\$120	200.000	(x accel, mm/sec^2)
\$121	200.000	(y accel, mm/sec^2)
\$122	10.000	(z accel, mm/sec^2)
\$130	200.000	(x max travel, mm)
\$131	200.000	(y max travel, mm)
\$132	200.000	(z max travel, mm)

Fonte: Próprio autor

Na adaptação do aplicativo utilizando-se o Android Studio onde adicionou funcionalidade de comunicar via Bluetooth baseado num código disponível no GitHub com nome *bluetooth-spp-terminal*, onde aperta-se um botão ligar /desligar na parte de menu do aplicativo, e ele se conecta diretamente com módulo Bluetooth, pois o endereço MAC foi inserido no código evitando de ficar procurando as conexões Bluetooth disponíveis. Os comandos G-code são gerados de maneira que haja uma concatenação de *Strings* com as coordenadas do desenho de bordado

onde é armazenado uma *array* para depois ser enviando para o Arduino. Também adicionado botões como start, stop, reset, destravar, *homing*, mais(+), menos(-), e os direcionais +x, -x, +y e -y na tela.

- **Start:** inicia o envio dos comandos para o Arduino do desenho selecionado.
- **Stop:** para a máquina.
- **Reset:** volta para centro do desenho.
- **Desbloquear:** como habilitado a função *homing* o Arduino fica travado esperando ser enviado o comando \$X para poder responder aos outros comandos.
- **Homing:** envia comandos para que ele centralize o bastidor na máquina de costura em relação agulha.
- **Mais(+) e Menos(-):** tem a função de avançar e recuar pontos.
- **Direcionais +x, -x, +y e -y:** envia comandos para posicionar o bastidor no ponto desejado para bordar.

Figura 16 – Aplicativo modificado



Fonte: Próprio autor

Gastos necessários para desenvolver o sistema segue na tabela 3 abaixo.

Tabela 3 – Gastos do projeto

Unidades	Materiais	Preço(R\$)
1	Chapa de ferro galvanizado	Grátis
1	Perfil de Alumínio 20x40x100cm	34,90

5	Par de rodanas com rolamento	99,50
2	Polia GT2 16 dentes	10,00
1	Correia Gt2 6mm 1m	18,90
4	Chave de fim de curso	6,40
1	Kit cnc shield , Ardino Uno e 4 drives A4988	70,00
1	Modulo Bluetooth HC – 05	16,90
2	Motor de Passo Nema 17	137,80
25	Parafuso em geral	20,00
1	Modulo Relé	Grátis
1	Máquina de costura	Grátis
TOTAL		414,40

Fonte: Próprio autor

Na tabela podemos observar o custo e fazer um comparativo principais máquina de bordado doméstica com funções básicas do mercado, são máquinas Brother PE770 com custo de R\$3216,02 e máquina Janome Memory Craft 230E com um custo de R\$3190,00.

Podemos fazer uma tabela comparativa logo abaixo onde foi atribuído uma nota de 0 a 5 para requisitos comprados entre as máquinas citadas acima e com a máquina do projeto.

Tabela 4 – Comparativo entre máquinas de bordado com a do projeto

Requisitos	Máquina do projeto	Brother PE770	Janome Memory Craft 230E
Qualidade do tipos de pontos	3	5	5
Visualizar bordado	5	2	2
Velocidade para confecção do bordado	3	5	5
Recurso adicionais	2	4	4

Fonte: Próprio autor

A notas atribuídas para máquina da Brother e para da Janome foram iguais pois suas características são muito equivalentes.

Na comparação dos tipos de pontos que são satin e tatami e ponto reto a máquina do projeto não ficou com ponto satin e tatami similar as demais, mas já o ponto reto ficou similar.

Já a visualização dos bordados das maquina Brother PE770 e Janome Memory Craft 230E fica a desejar pois são monocromáticas e muito pequenas sendo muito difícil de identificar o desenho selecionado por outro lado a máquina do projeto possui uma ótima visualização.

O tempo de confecção de desenho de bordado da máquina de projeto mostrou ser inferior as máquinas aqui citadas, devido possuir uma velocidade constante da mesa.

Nos recursos adicionais são aqueles na qual a máquina tem a capacidade de posicionar, girar, espelhar, aumentar e reduzir o desenho e com fontes de letras pré-carregadas na máquina onde é possível escrever nomes. A máquina do projeto não possui esses requisitos apenas em posicionar o desenho no bastidor. As máquinas aqui citadas receberam nota 4 pela dificuldade de operar tendo uma visualização não tão boa do desenho no display.

Na figura 17 apresenta-se o produto final onde o aplicativo Android interpreta o desenho de bordado envia para máquina do projeto resultando o bordado.

Figura 17 – Produto final



Fonte: Próprio autor

5.Considerações finais

Obteve-se resultados satisfatórios no objetivo de automatizar a máquina de costura convencional para converte-la em uma máquina de bordados a um baixo custo, tendo uma eficiência pouco abaixo das principais máquinas de bordado doméstica do mercado, devido possuir velocidades constantes tanto da máquina de costura quanto a mesa, dependendo da complexibilidade do desenho o tempo gasto para conclusão é maior e a qualidade do bordado é inferior.

Para possuir um bordado similar a essas máquinas de bordados necessitaria de melhoria do *software* de interface do usuário responsável pela leitura, pois estes arquivos são próprios para máquinas de bordados onde se pode tirar mais informações desses arquivos, como o tipo de ponto ou até mesmo os comandos para ser enviados para máquina, diferentemente do que foi

usado só as coordenadas, assim a velocidade da mesa estaria variando a partir de cada tipo de ponto. Também seria necessário controlar a velocidade de máquina de costura, propondo um controle de potência CA (Corrente Alternada) por PWM (*Pulse Width Modulation*) controlado pelo Arduino, assim conseguiríamos um controle de velocidade da máquina de costura pelos tipos de pontos identificado no aplicativo, consequentemente buscando um bordado de mais qualidade.

REFERÊNCIAS

ARDUINO. **Arduino Uno Rev3**. 2018. Disponível em: < <https://store.arduino.cc/usa/arduino-uno-rev3> >. Acesso em: 23 de setembro de 2018.

ARDUINO E CIA. **Como usar o driver A4988 como motor de passo Nema 17**. 2015. Disponível em:<<https://www.arduinoecia.com.br/2015/03/driver-a4988-com-motor-de-passo-nema-17.html>>. Acesso em: 11 de setembro de 2018.

EVANS, Martin; NOBLE , Joshua; HOCHENBAUM, Jordan. **Arduino em Ação**. 1. ed. São Paulo: Novatec, 2013. 424 p.

GITHUB. **Mobile Embroidery Viewer**. Disponível em:< <https://github.com/Embroidermodder/MobileViewer> >. Acesso em: 23 de abril de 2018.

GITHUB. **bluetooth-spp-terminal**. Disponível em:< <https://github.com/Sash0k/bluetooth-spp-terminal>>. Acesso em: 26 de junho de 2018.

LEAL, Nelson Glauber de Vasconcelos. **Dominando o Android do básico ao avançado**. 2. ed. São Paulo: Novatec, 2016. 951 p.

LEITE, Josicleudo R et al. Desenvolvimento de um cnc de pequeno porte. **Reunião Regional da Sbpcc**, Cariri-CE, p.1-3, 2 maio 2017. Disponível em: <<http://www.sbpccnet.org.br/livro/cariri/resumos/2238.pdf>>. Acesso em: 16 ago. 2018.

MCCROBERTS, Michael. **Arduino básico**. São Paulo, SP: Novatec, 2011. 453 p.

MONK, Simon. **Projetos com Arduino e Android**: use seu smartphone ou tablet para controlar o Arduino. Porto Alegre: Bookman, 2014. 212 p.

OPENBUILDS. **Embroidery machine with XY belt and pinion drive**. 2014. Disponível em:<<https://openbuilds.com/builds/embroidery-machine-with-xy-belt-and-pinion-drive.691/>>. Acesso em: 14 de maio de 2018.

ROSÁRIO, João Maurício. **Princípios de Mecatrônica**. 1. ed. São Paulo: Prentice Hall, 2005. 356 p.

SALLES, Fábio Moreira et al. Desenvolvimento de uma fresado cnc de três eixos do tipo router. **Jornada de Iniciação Científica da Faacz**, [S. l.], p.1-4, 2017. Disponível em: <http://www.faacz.com.br/portal/conteudo/iniciacao_cientifica/programa_de_iniciacao_cientifica/2017/anais/desenvolvimento_de_uma_fresado_cnc_de_tres_eixos_do_tipo_router.pdf>. Acesso em: 16 ago. 2018.

SANTI, Kelvin R. C.; SPEROTTO, Lucas K.. Desenvolvimento De Fresadora CNC de Baixo Custo. **Anais da Escola Regional de Informática da Sociedade Brasileira de Computação (SBC) – Regional de Mato Grosso**, [S.l.], v. 7, dez. 2016. ISSN 2447-5386. Disponível em: <<http://anaiserimt.ic.ufmt.br/index.php/erimt/article/view/89>>. Acesso em: 16 ago. 2018.

SILVA, Sidnei Domingues da. **CNC: Programação De Comandos Numéricos Computador**. 1. ed. São Paulo: Erica, 2002. 312 p.

STEVAN JR., Sergio Luiz; SILVA, Rodrigo Adamshuk. **Automação e instrumentação industrial com arduino: teoria e projetos**. São Paulo, SP: Saraiva: Erica, 2015. 296 p.

SUH, Suk-Hwan, et al. **Theory and Design of CNC Systems**. 1. ed. Espanha: Springer, 2008. 455 p.

VARGHESE, Leeba et al. Experimental Study using a Low Cost and Versatile 3-Axis CNC Router with Bluetooth Connectivity. **Journal Of Advanced Engineering Research**. [S. l.], p. 20-29. 2017. Disponível em: <https://dl.jaeronline.com/papers/vol04issue012017/05_JAER_V4N1.pdf>. Acesso em: 16 ago. 2018.