

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Rodolfo Mares Malta

Controlador MIDI baseado em Arduino

Uberlândia, Brasil

2018

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Rodolfo Mares Malta

Controlador MIDI baseado em Arduino

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Msc. William Chaves de Souza Carvalho

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2018

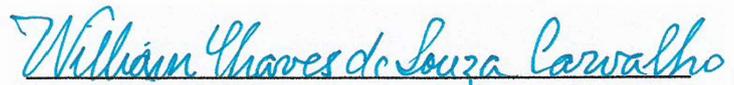


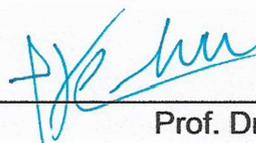
Ata de Defesa do Trabalho de Conclusão de Curso

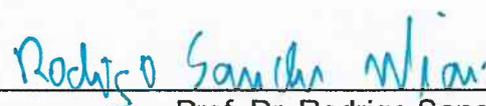
Ata da 1ª defesa de trabalho de conclusão do curso de Bacharelado em Sistemas de Informação, realizada em 17/12/2018, na sala 1B230 do Campus Santa Mônica, às 18:30 pelo aluno **Rodolfo Mares Malta**, Matrícula 11511BSI223. A banca examinadora reuniu-se para avaliar o trabalho de conclusão de curso intitulado **Controlador MIDI baseado em Arduino**. Em sessão pública com cerca de 50 minutos de duração, o aluno expôs seu trabalho e foi arguido oralmente pelos membros da banca, sendo considerado aprovado (aprovado/reprovado) com nota final 100.

Uberlândia, 17 de dezembro de 2018

Banca examinadora:


Prof. Msc. William Chaves de Souza Carvalho
(Orientador)


Prof. Dr. Márcio José da Cunha


Prof. Dr. Rodrigo Sanches Miani

Agradecimentos

À minha namorada, pelo incentivo, apoio, paciência, compreensão e pela grande ajuda com a logística no processo de construção deste trabalho.

À minha mãe, por estar sempre comigo nos momentos mais difíceis da vida.

Aos amigos que estiveram ao meu lado e contribuíram com suas opiniões e palavras de incentivo.

Ao professor William, por abraçar a ideia e me orientar durante o desenvolvimento deste trabalho.

Resumo

O protocolo MIDI, proposto no início dos anos 1980 como forma de padronizar a comunicação entre diferentes dispositivos musicais, atualmente é utilizado em diversos instrumentos musicais eletrônicos e softwares de produção musical e discotecagem, do mais simples ao mais sofisticado. Este trabalho apresenta detalhes teóricos e técnicos do projeto e construção de um dispositivo eletrônico capaz de controlar qualquer software compatível com MIDI sem a necessidade de drivers ou softwares adicionais. A plataforma Arduino foi utilizada pela facilidade de se encontrar informações relacionadas ao seu uso e também pelo baixo custo. Os objetivos estabelecidos inicialmente foram atingidos e, por fim, mostrou-se viável a construção de um controlador MIDI personalizado de baixo custo, baseado em Arduino e compatível com diversos softwares e sistemas operacionais.

Palavras-chave: Controlador MIDI, Arduino, Multiplataforma, *MIDI Class Compliant*.

Lista de ilustrações

Figura 1 – Esquemático do multiplexador CD4051B. Fonte: (TEXAS INSTRUMENTS, 1998)	13
Figura 2 – Traktorino. Fonte: (Arduino Blog, 2018)	14
Figura 3 – Rascunho da ideia inicial. Fonte: autoria própria	17
Figura 4 – Desenho final em CAD, reduzido. Fonte: autoria própria	18
Figura 5 – Projeto da PCI em duas camadas. Fonte: autoria própria	19
Figura 6 – Desenho da placa impresso em papel transfer. Fonte: autoria própria	20
Figura 7 – Primeira etapa da soldagem da PCI. Fonte: autoria própria	22
Figura 8 – Montagem dos componentes na chapa de acrílico. Fonte: autoria própria	23
Figura 9 – Montagem final. Fonte: autoria própria	23
Figura 10 – MIDI Monitor. Fonte: autoria própria	26
Figura 11 – Tela principal do Traktor Pro 3. Fonte: autoria própria	27
Figura 12 – Traktor Pro com área do <i>browser</i> expandida. Fonte: autoria própria	28
Figura 13 – Caustic 3 - linha do tempo. Fonte: autoria própria	29
Figura 14 – Caustic 3 - BeatBox. Fonte: autoria própria	29
Figura 15 – Anúncio de venda do Traktor Kontrol Z1. Fonte: (MERCADOLIVRE, 2018)	31
Figura 16 – Anúncio de venda do Hercules DJ Control Compact. Fonte: (MERCADOLIVRE, 2018)	31

Lista de abreviaturas e siglas

CAD	<i>Computer Aided Design</i>
CV	<i>Control voltage</i>
DAW	<i>Digital Audio Workstation</i>
DIY	<i>Do It Yourself</i>
EDA	<i>Electronic Design Automation</i>
IDE	<i>Integrated Development Environment</i>
LED	<i>Light Emitting Diode</i>
LGPL	<i>GNU Lesser General Public License</i>
MIDI	<i>Musical Instrument Digital Interface</i>
MIT	<i>Massachusetts Institute of Technology</i>
MMA	<i>MIDI Manufacturers Association</i>
PCI	Placa de Circuito Impresso
SMD	<i>Surface-mount Device</i>
TMA	<i>The MIDI Association</i>
USB	<i>Universal Serial Bus</i>

Sumário

1	INTRODUÇÃO	9
1.1	Objetivos	10
1.2	Metodologia	10
1.3	Organização do Trabalho	10
2	REVISÃO BIBLIOGRÁFICA	11
2.1	Protocolo MIDI	11
2.1.1	Mensagens MIDI	11
2.2	Arduino	12
2.3	Eletrônica digital	12
2.3.1	Multiplexador	12
2.4	Trabalhos Correlatos	14
3	DESENVOLVIMENTO	15
3.1	Projeto de Hardware	15
3.1.1	Definição dos componentes	15
3.1.2	Desenho e construção	16
3.1.3	Placa de Circuito Impresso (PCI)	18
3.1.3.1	Problema encontrado	18
3.1.3.2	Solução: PCI de face única	19
3.1.3.3	Confecção da placa	19
3.1.4	Montagem	21
3.2	Projeto de Software	24
3.2.1	O protocolo de comunicação	24
3.2.2	Leitura das entradas	24
3.2.2.1	Entradas digitais: botões	24
3.2.2.2	Entradas analógicas: potenciômetros	25
3.2.3	Função “shift”	25
3.3	Testes	25
3.4	Integração com software compatível com o protocolo MIDI	26
3.4.1	Traktor Pro	26
3.4.2	Caustic	28
4	CONSIDERAÇÕES FINAIS	30
	Considerações Finais	30

4.1	Custos do projeto	30
4.1.1	Comparativo de preços de produtos similares	30
4.1.1.1	Traktor Kontrol Z1	31
4.1.1.2	Hercules DJ Control Compact	31
4.2	Dificuldades e Desafios	32
4.3	Trabalhos Futuros	32
4.4	Conclusão	32
	 REFERÊNCIAS	 34

1 Introdução

O padrão MIDI - *Musical Instrument Digital Interface* foi proposto no início dos anos 1980 por um grupo de fabricantes de sintetizadores como forma de padronizar a comunicação entre diferentes instrumentos musicais (MANCINI, 2018).

Até o final dos anos 1970 era comum a utilização de sintetizadores modulares analógicos que possuíam um sistema de controle também analógico conhecido por *control voltage* (CV). O sistema consistia de cabos ligando um sintetizador (que agiria como um controlador) a um ou mais sintetizadores, que teriam seus osciladores controlados por este primeiro. Os osciladores eram controlados por uma tensão aplicada a esse cabo de controle. Uma alteração na tensão, gerada pelo teclado do sintetizador controlador, implicava em uma alteração nos parâmetros que estavam sendo controlados, nesse caso notas musicais. Cada nota tinha uma tensão correspondente, e alterações da tensão em 1 volt resultavam em uma mudança de oitava (HUBER, 2007).

Esse sistema, embora pouco sofisticado, funcionava para os sintetizadores monofônicos (que reproduzem apenas um som de cada vez) da época. Porém, deixou de ser eficaz com o surgimento dos sintetizadores polifônicos (capazes de reproduzir mais de um som ao mesmo tempo). Por exemplo, para controlar um sintetizador polifônico utilizando CV seria necessário utilizar um cabo para cada tecla presente no teclado, solução pouco prática e, portanto, inviável.

A integração entre dispositivos de diferentes fabricantes também era um ponto negativo do sistema *control voltage*. Como não se tratava de um padrão, cada fabricante desenvolvia sua própria forma de comunicação, o que tornava necessária a utilização de dispositivos especiais para intermediar a conexão entre dispositivos de fabricantes diferentes. (MANCINI, 2018).

A grande necessidade de padronização, aliada aos avanços tecnológicos que permitiram a construção de circuitos eletrônicos a um custo mais baixo, levou à criação e aceitação do protocolo MIDI pela indústria (HUBER, 2007). Hoje pode ser encontrado em diversos instrumentos musicais eletrônicos e equipamentos relacionados, dos mais simples aos mais sofisticados (ANDERTON, 2013).

Apesar de existir uma grande variedade de controladores MIDI disponíveis no mercado atualmente, a possibilidade de produzir dispositivos eletrônicos utilizando microcontroladores permite que pessoas com conhecimentos básicos de eletrônica e programação construam seus próprios equipamentos MIDI de forma consideravelmente mais barata, sendo esta a principal motivação para o desenvolvimento deste trabalho.

1.1 Objetivos

O objetivo geral deste trabalho é construir um dispositivo eletrônico de baixo custo capaz de controlar qualquer software DAW (*Digital Audio Workstation*, em português Estação de Trabalho de Áudio Digital) compatível com o protocolo MIDI, de forma que seja possível atribuir controles físicos a determinadas funções do software.

1.2 Metodologia

- Utilizar conceitos de eletrônica digital para projetar o dispositivo
- Estudar a utilização de softwares EDA (do inglês: *Electronic Design Automation*) para criação de placas de circuito impresso
- Confeccionar a placa de circuito impresso
- Utilizar software CAD (Desenho Assistido por Computador, do inglês: *Computer Aided Design*) para produzir o corte automatizado das chapas de acrílico e aço utilizadas no acabamento
- Programar um microcontrolador, especificamente o Arduino, para que atue como um dispositivo MIDI USB e seja compatível com qualquer software que suporte este protocolo sem a necessidade de *drivers* ou softwares intermediários.

1.3 Organização do Trabalho

O capítulo 2 aborda conceitos relacionados às tecnologias utilizadas na construção deste trabalho. O capítulo 3 conterà o desenvolvimento, detalhando o processo de construção, materiais utilizados e testes. O capítulo 4 conterà os resultados obtidos, custos do projeto, dificuldades e conclusão.

2 Revisão Bibliográfica

Este capítulo apresenta os principais conceitos necessários para o desenvolvimento deste trabalho, bem como alguns trabalhos correlatos.

2.1 Protocolo MIDI

A especificação do protocolo MIDI é definida pela MIDI Manufacturers Association (MMA), uma associação comercial sem fins lucrativos de empresas que desenvolvem produtos que utilizam este protocolo ([THE MIDI ASSOCIATION, 2018](#)).

Uma das principais fontes de informações relacionadas ao protocolo MIDI é o site da The MIDI Association (TMA) ([THE MIDI ASSOCIATION, 2018](#)), comunidade administrada pela MMA aberta a pessoas que utilizam desta tecnologia e desejam compartilhar seus conhecimentos e experiências, como músicos, DJs, engenheiros de áudio, entusiastas DIY (“faça você mesmo”, do inglês: *Do It Yourself*), entre outros.

O documento *The Complete MIDI 1.0 Specification* ([MIDI MANUFACTURERS ASSOCIATION, 2014](#)), disponível no site da The MIDI Association, fornece um “retrato” de como era definido o protocolo MIDI em 1996. Apesar do nome sugestivo, este documento não contempla uma definição completa das especificações, visto que alguns complementos e adendos foram criados após 1996 ([THE MIDI ASSOCIATION, 2018](#)).

2.1.1 Mensagens MIDI

A comunicação entre os dispositivos MIDI é feita através de mensagens (*MIDI Messages*). De acordo com a especificação oficial do protocolo ([MIDI MANUFACTURERS ASSOCIATION, 2014](#)), uma mensagem MIDI é composta de 8 bits, podendo vir acompanhada de 2 bits extras indicando início e fim.

Existem vários tipos de mensagens MIDI, que são usadas para diferentes propósitos. Para o desenvolvimento deste trabalho foram utilizadas as mensagens do tipo *note on*, *note off* e *control change*, detalhadas a seguir:

- ***note on***: Indica que uma tecla foi pressionada no teclado ou controlador MIDI, sinalizando que determinada nota deve ser reproduzida
- ***note off***: Indica que a tecla foi liberada (não está mais sendo pressionada). Caso não seja enviada, a nota continua sendo reproduzida infinitamente

- **control change:** Normalmente utilizada para transmitir informações sobre a posição de controles giratórios ou deslizantes, *encoders* ou mesmo pedais.

2.2 Arduino

Arduino é uma plataforma *open source* de prototipagem eletrônica, de hardware e software livres e extensíveis (ARDUINO, 2018a). Existem várias versões produzidas pelo fabricante oficial mas, por se tratar de uma plataforma livre, é fácil encontrar placas modificadas produzidas por outros fabricantes, as chamadas “Arduino compatíveis”. As placas compatíveis podem ter recursos modificados, mas mantém a compatibilidade com a IDE do Arduino e podem ser programadas da mesma forma que um Arduino original.

Dentre os motivos que justificam a escolha dessa plataforma para desenvolvimento deste trabalho os principais são a facilidade de uso, possibilidade de desenvolver desde projetos simples aos mais avançados, ambiente de desenvolvimento (IDE, do inglês: *Integrated Development Environment*) compatível com os principais sistemas operacionais atuais (Linux, Mac OS X, e Windows) (ARDUINO, 2018a) e facilidade de encontrar informações relacionadas ao seu uso, através do fórum mantido pela comunidade de usuários em todo o mundo (ARDUINO, 2018b).

A versão utilizada neste trabalho é o Arduino Pro Micro, uma versão não oficial comercializada inicialmente pela SparkFun. Esta versão conta com microcontrolador ATmega32u4, o mesmo utilizado na placa oficial Arduino Leonardo, cristal de 16MHz, conexão micro USB, 12 portas digitais de entrada/saída e 4 portas analógicas.

2.3 Eletrônica digital

O desenvolvimento deste trabalho baseou-se em alguns conceitos de eletrônica digital para o projeto do circuito. Dentre os componentes eletrônicos utilizados, merece destaque o circuito integrado multiplexador, descrito a seguir.

2.3.1 Multiplexador

Um multiplexador, também conhecido como MUX, é um dispositivo eletrônico que permite combinar diversas fontes de entrada de dados em uma única saída (FLOYD, 2007). Através de suas entradas seletoras é possível indicar qual das entradas de dados será lida em determinado instante. Por este motivo os multiplexadores também são conhecidos como seletores de dados, pois funcionam como uma chave de seleção entre as várias entradas conectadas.

Um multiplexador também pode atuar como um demultiplexador, fazendo o processo inverso: combinando uma única entrada de dados para várias saídas diferentes, porém essa função não foi necessária para o desenvolvimento deste projeto.

O modelo utilizado foi o CD4051B, fabricado pela Texas Instruments. Este componente possui 8 entradas de dados e 3 entradas seletoras. No *datasheet* é possível encontrar todas as suas especificações técnicas, como a Figura 1, que mostra a posição de cada pino no componente. Suas funções são:

- **CHANNELS IN/OUT:** Múltiplas entradas de dados (ou saídas, caso usado como demultiplexador)
- **COM OUT/IN:** Saída única (ou entrada, caso usado como demultiplexador)
- **INH:** Pino *Enable*. Desliga os canais do multiplexador caso receba um sinal de nível lógico alto. Este pino foi ligado diretamente ao terra.
- **V_{EE}** Entrada de energia, pino negativo
- **V_{SS}** *Ground* (terra)
- **V_{DD}** Entrada de energia, pino positivo
- **A, B, C** Entradas seletoras.

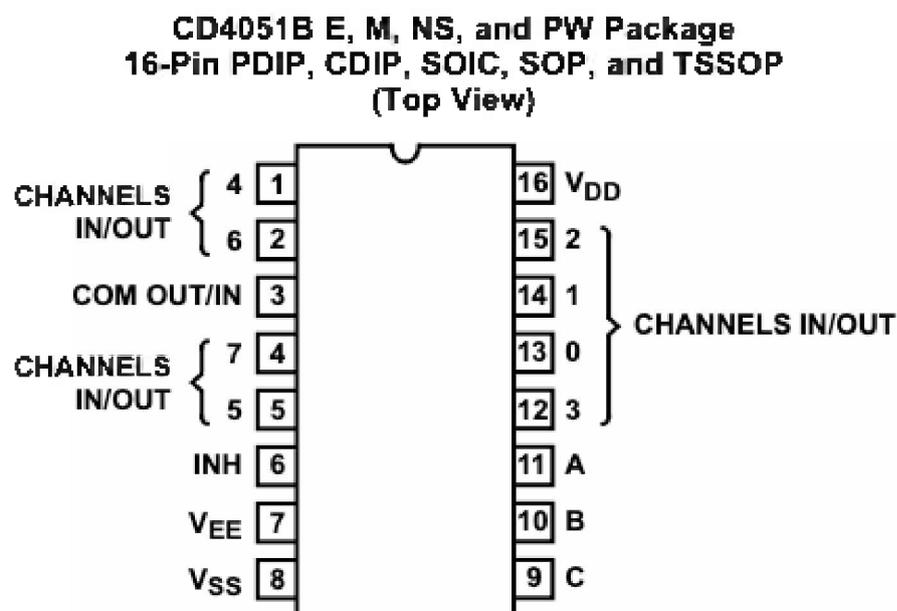


Figura 1 – Esquemático do multiplexador CD4051B. Fonte: (TEXAS INSTRUMENTS, 1998)

2.4 Trabalhos Correlatos

HIDUINO (HIDUINO, 2011) é um projeto *open source* disponível no GitHub sob licença MIT (*Massachusetts Institute of Technology*). Consiste em um *firmware* capaz de transformar um dispositivo Arduino UNO ou Arduino Mega 2560 em um dispositivo MIDI USB nativo. Por padrão o Arduino é reconhecido pelo sistema operacional como um dispositivo de comunicação serial, sendo necessário um software intermediário, como o Hairless MIDI<->Serial Bridge (GRATTON, 2011) para transformar as mensagens recebidas pela porta serial em mensagens MIDI. Com o HIDUINO elimina-se a necessidade de softwares intermediários, pois o controlador será reconhecido pelo sistema operacional como um dispositivo de interface humana (HID, do inglês: *Human Interface Device*).

MIDIUSB Library for Arduino (GREWAL, 2015) é uma biblioteca *open source* (licença LGPL - *GNU Lesser General Public License*) que, assim como o HIDUINO, permite transformar um Arduino em um dispositivo MIDI USB. Porém, oferece suporte outros modelos de Arduino, como Leonardo, Micro, Due, Zero, entre outros.

Na Internet é possível encontrar diversos exemplos de controladores MIDI personalizados, como por exemplo o Traktorino, um dispositivo *open source* desenvolvido por Gustavo Silveira, do site Nerd Musician (SILVEIRA, 2018). A Figura 2 mostra os estágios de sua montagem.

O Traktorino foi construído com o mesmo propósito do projeto descrito neste trabalho, porém diferencia-se por não ter a possibilidade de mapear múltiplos controles a uma mesma tecla, função esta que recebeu o nome de “shift” e será detalhada no capítulo de desenvolvimento.

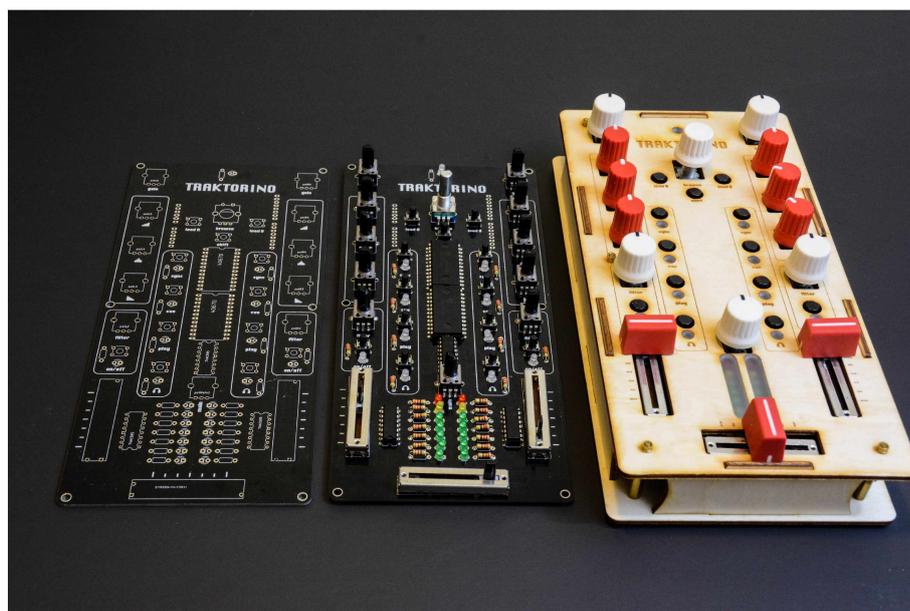


Figura 2 – Traktorino. Fonte: (Arduino Blog, 2018)

3 Desenvolvimento

Este capítulo descreve as etapas do desenvolvimento deste trabalho, divididas entre projeto de hardware, de software, e testes.

3.1 Projeto de Hardware

3.1.1 Definição dos componentes

O primeiro passo do desenvolvimento deste trabalho foi definir os componentes eletrônicos a serem utilizados, dos quais o mais importante é o Arduino. A escolha do modelo Pro Micro se deu pelo fato de ser pequeno, de preço relativamente barato, e pela possibilidade de transformá-lo em um dispositivo *MIDI Class Compliant*. Portanto após programado, quando ligado a uma porta USB, o computador (ou mesmo um smartphone, tablet, etc) o reconhecerá como dispositivo MIDI sem necessidade de instalação de *drivers* ou *softwares* intermediários para isso.

Apesar de não ser produzido e comercializado pelo fabricante original do Arduino, o Pro Micro utiliza o mesmo microcontrolador do Arduino Micro, Leonardo e outros, o ATmega32U4, e é totalmente compatível com a Arduino IDE.

Componentes eletrônicos utilizados:

- 1 Arduino Pro Micro
- 16 potenciômetros giratórios lineares 10k Ω
- 16 *knobs* para potenciômetros giratórios de eixo estriado
- 4 potenciômetros deslizantes lineares 10k Ω , 45mm de curso
- 4 *knobs* para potenciômetros deslizantes (opcional)
- 13 botões momentâneos (*push buttons*)
- 4 multiplexadores CD4051B
- 4 LEDs
- 1 placa de fenolite face simples
- Fios para a conexão dos componentes à placa de circuito

Componentes Estruturais:

- Chapa de acrílico branco translúcido, 3mm de espessura
- Chapa de aço inox escovado, 1mm de espessura
- Sarrafo de madeira, 15mm de espessura
- Parafusos, porcas e arruelas
- Fita dupla face
- Cola para madeira

Ferramentas utilizadas:

- Furadeira
- Ferro de passar
- Chave philips
- Estilete
- Alicates de corte
- Ferro de solda 30W
- Estanho e pasta para soldar
- Pistola de cola quente
- Serra elétrica
- Máquinas de corte a *laser* e jato d'água (serviço terceirizado)

3.1.2 Desenho e construção

Após a definição dos componentes eletrônicos, foi feito o desenho da disposição desses componentes.

Por se tratar de um dispositivo que poderá ser utilizado por um DJ durante apresentações ao vivo, é importante que os controles físicos sejam visíveis em ambientes escuros. Para isso, a parte superior foi construída em acrílico branco translúcido, de forma que os LEDs instalados na parte interna iluminem os controles. Os potenciômetros e botões foram fixados a esta chapa de acrílico.

Para o acabamento, optou-se por uma chapa de aço inox escovado logo acima do acrílico. Este material foi escolhido por questões estéticas e por ser rígido, protegendo o acrílico e aumentando sua resistência.

As peças foram desenhadas no software AutoCAD. Com este desenho foi possível realizar os cortes de forma precisa em máquinas de corte a *laser* (para o acrílico), e a jato d'água (para a chapa de aço). A exceção foi a caixa de madeira, construída manualmente.

Até chegar ao desenho final foram feitos vários rascunhos (Figura 3) em papel e papelão, para ter uma noção da melhor distância entre os potenciômetros e botões visando uma utilização prática e ergonômica dos controles. Após finalizado (Figura 4), para eliminar qualquer chance de erro de projeto, o desenho foi impresso em tamanho real em papel A3 para conferência antes de ser enviado para o serviço de corte dos materiais definitivos (acrílico e aço).

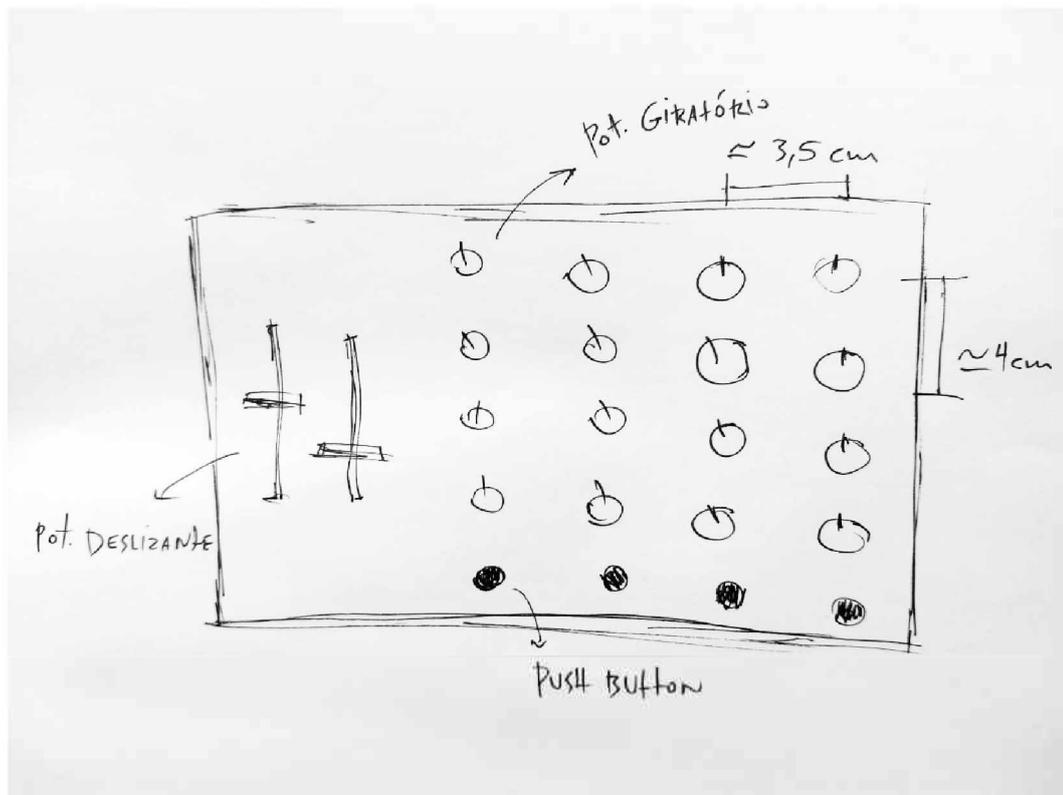


Figura 3 – Rascunho da ideia inicial. Fonte: autoria própria

A caixa de madeira foi a última parte a ser construída, pois precisava ser feita na medida exata para acomodar os componentes internos após a soldagem dos fios na placa de circuito.

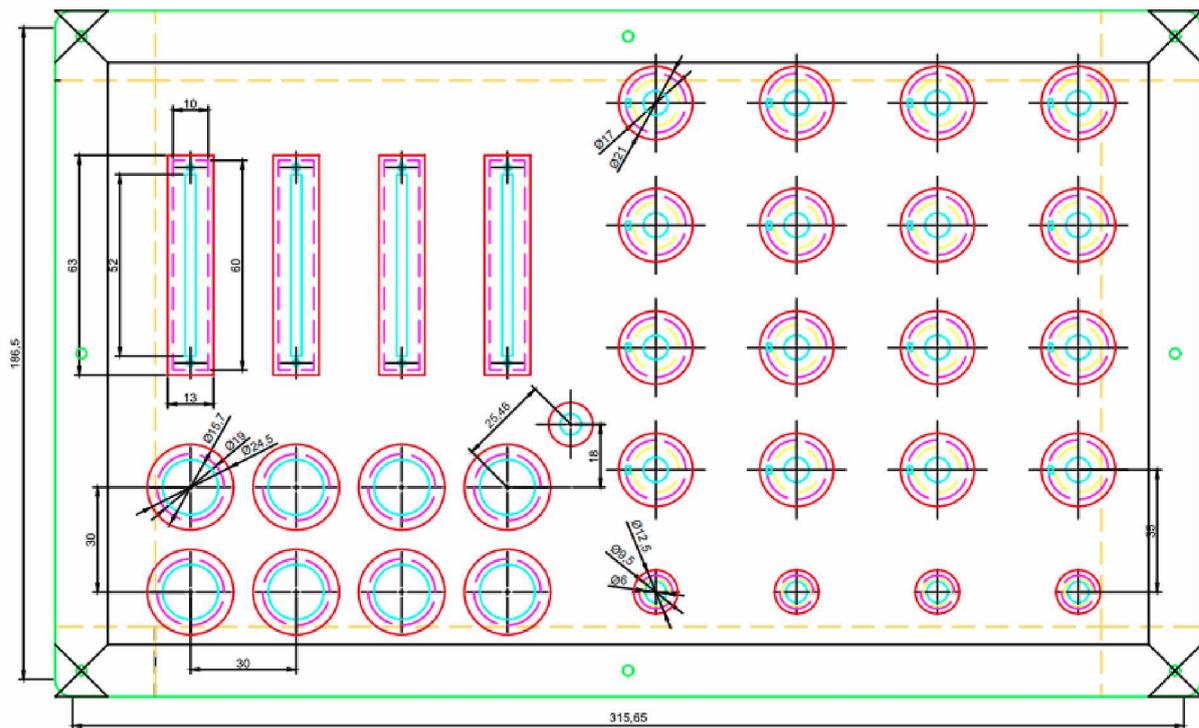


Figura 4 – Desenho final em CAD, reduzido. Fonte: autoria própria

3.1.3 Placa de Circuito Impresso (PCI)

Visando reduzir a quantidade de fios e melhorar a qualidade das conexões elétricas dos componentes, foi projetada uma placa de circuito impresso utilizando o software Eagle. O projeto envolve primeiramente a criação do esquemático com as conexões elétricas entre os componentes, e então a placa em si.

3.1.3.1 Problema encontrado

Devido à quantidade de trilhas para as conexões entre o Arduino e os 4 multiplexadores, o projeto resultou em uma placa de duas camadas. Isso seria necessário para que as trilhas não se cruzassem, ocasionando um curto-circuito. A Figura 5 mostra o projeto da placa em duas camadas. As linhas azuis indicam a camada inferior, as linhas vermelhas indicam a camada superior, e os pontos representam as vias, que são furos metalizados interligando as duas camadas. Os *pads* são as terminações onde seriam soldados os fios para ligar os potenciômetros e botões.

O problema da placa de dupla face é que os furos para soldagem dos componentes precisam ser metalizados, para conectar as trilhas da face superior às trilhas da face inferior, porém o processo de metalização dos furos mostrou-se inviável de se fazer utilizando métodos caseiros.

Existem empresas especializadas em confecção de PCI, mas o custo de produção

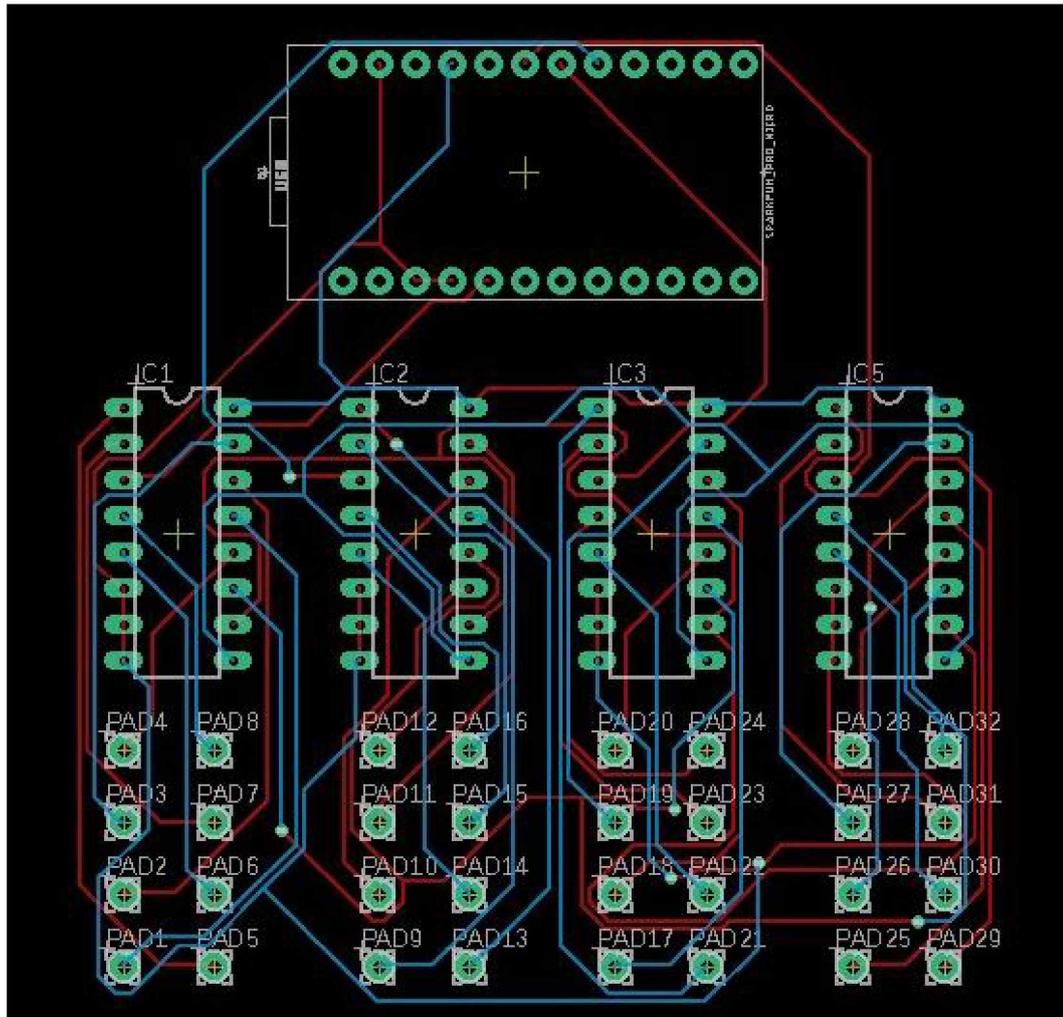


Figura 5 – Projeto da PCI em duas camadas. Fonte: autoria própria

inviabilizaria completamente este projeto. Por exemplo, em uma das cotações a quantidade mínima de placas para a fabricação era 5 unidades, ou valor mínimo de R\$ 500,00. Portanto, foi feita uma adaptação.

3.1.3.2 Solução: PCI de face única

A solução encontrada foi modificar o desenho de forma que fosse possível utilizar apenas um lado da placa (face inferior) para fazer as trilhas de conexões entre os componentes. As conexões da face superior foram substituídas por *pads* que seriam conectados entre si através de *jumpers*, pequenos pedaços de fio interligando dois pontos na placa.

3.1.3.3 Confeção da placa

Para a confeção da placa foi utilizado o método de transferência térmica, que consiste em imprimir o desenho em um papel transfer utilizando uma impressora a laser

(Figura 6), e posteriormente aplicar este desenho à placa utilizando preferencialmente uma prensa térmica.

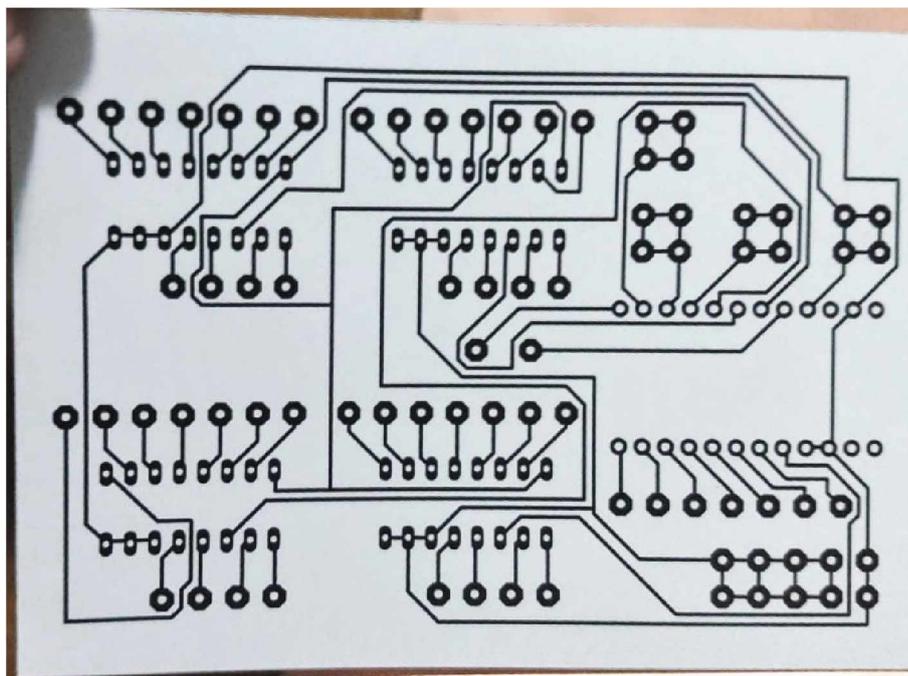


Figura 6 – Desenho da placa impresso em papel transfer. Fonte: autoria própria

Na falta de uma prensa térmica, utilizou-se um ferro de passar roupas, a seco. Após cortar a placa no tamanho exato e posicionar o papel transfer sobre a placa, o ferro de passar foi pressionado sobre o papel por aproximadamente 3 minutos em temperatura alta. Logo em seguida, a placa foi colocada sob água corrente para facilitar a retirada do papel que ficou grudado.

Como mostra a Figura ??, este processo faz o toner aderir à placa, criando uma proteção contra a corrosão do cobre, que é a próxima etapa do processo.

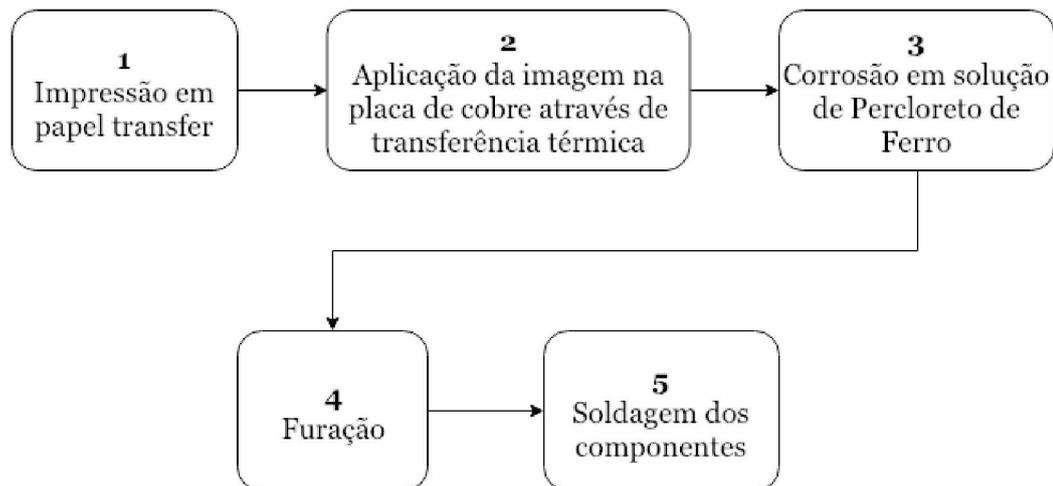
Após feita a transferência, a placa foi mergulhada em uma solução de perclorato de ferro e água, a uma concentração de 1 parte de perclorato para 1,5 partes de água. O perclorato de ferro é um composto químico que corrói toda a superfície de cobre da placa, exceto as partes em que a tinta foi aderida por transferência térmica. É comercializado em pó para diluição em água, e é facilmente encontrado em lojas de componentes eletrônicos.

O processo de corrosão levou cerca de 5 minutos. Após a corrosão a placa foi lavada em água corrente com uma esponja de lã de aço para remover a tinta, restando apenas as trilhas de cobre para interligar os componentes. Embora tenha sido utilizado um método caseiro, o resultado foi satisfatório.

A última etapa da confecção da placa de circuito foi a furação dos pontos de montagem dos componentes e dos *pads* para solda dos *jumpers* e dos fios de ligação dos potenciômetros e botões. Os furos foram feitos com uma furadeira e broca de 1mm de

espessura.

O diagrama a seguir detalha as etapas do processo de confecção da placa de circuito.



3.1.4 Montagem

O Arduino Pro Micro, possui apenas 12 pinos de I/O digitais, e 4 analógicos. Como o projeto previa mais controles do que portas disponíveis no Arduino, foram utilizados os 4 multiplexadores da seguinte forma:

- Um para os 8 botões de 19mm, ligado a uma entrada Digital do Arduino
- Dois para os 16 potenciômetros giratórios, ligados a duas entradas analógicas do Arduino
- Um deles, também ligado a uma entrada analógica, para os 4 potenciômetros deslizantes. Como o multiplexador CD4051B possui 8 entradas, 4 delas ficaram ociosas neste último.

Os 5 botões menores foram ligados diretamente nas portas digitais do Arduino.

A montagem foi iniciada pela soldagem dos componentes e *jumpers* na placa de circuito impresso, como mostra a Figura 7.

Enquanto aguardava a chegada da chapa de acrílico com os cortes a laser, os potenciômetros e botões foram montados em um pedaço de papelão cortado em tamanho real de acordo com o desenho impresso em papel A3, para que fosse feita a soldagem dos fios. Posteriormente os componentes foram movidos para a peça definitiva,

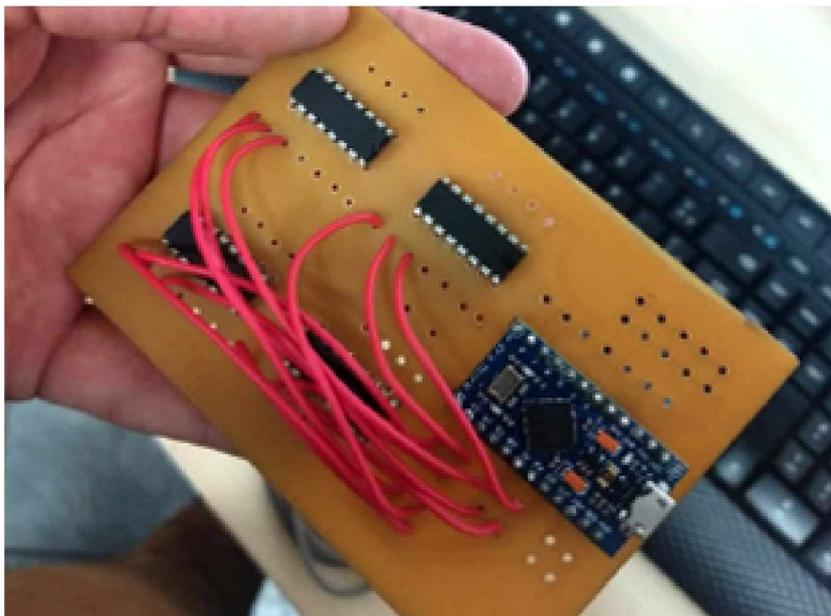


Figura 7 – Primeira etapa da soldagem da PCI. Fonte: autoria própria

em acrílico. A chapa de aço foi fixada à de acrílico com fita dupla face, e então ambas foram fixadas na caixa de madeira com parafusos.

A Figura 8 mostra os componentes já montados na chapa de acrílico e com os fios soldados na PCI, e a Figura 9 o projeto finalizado.

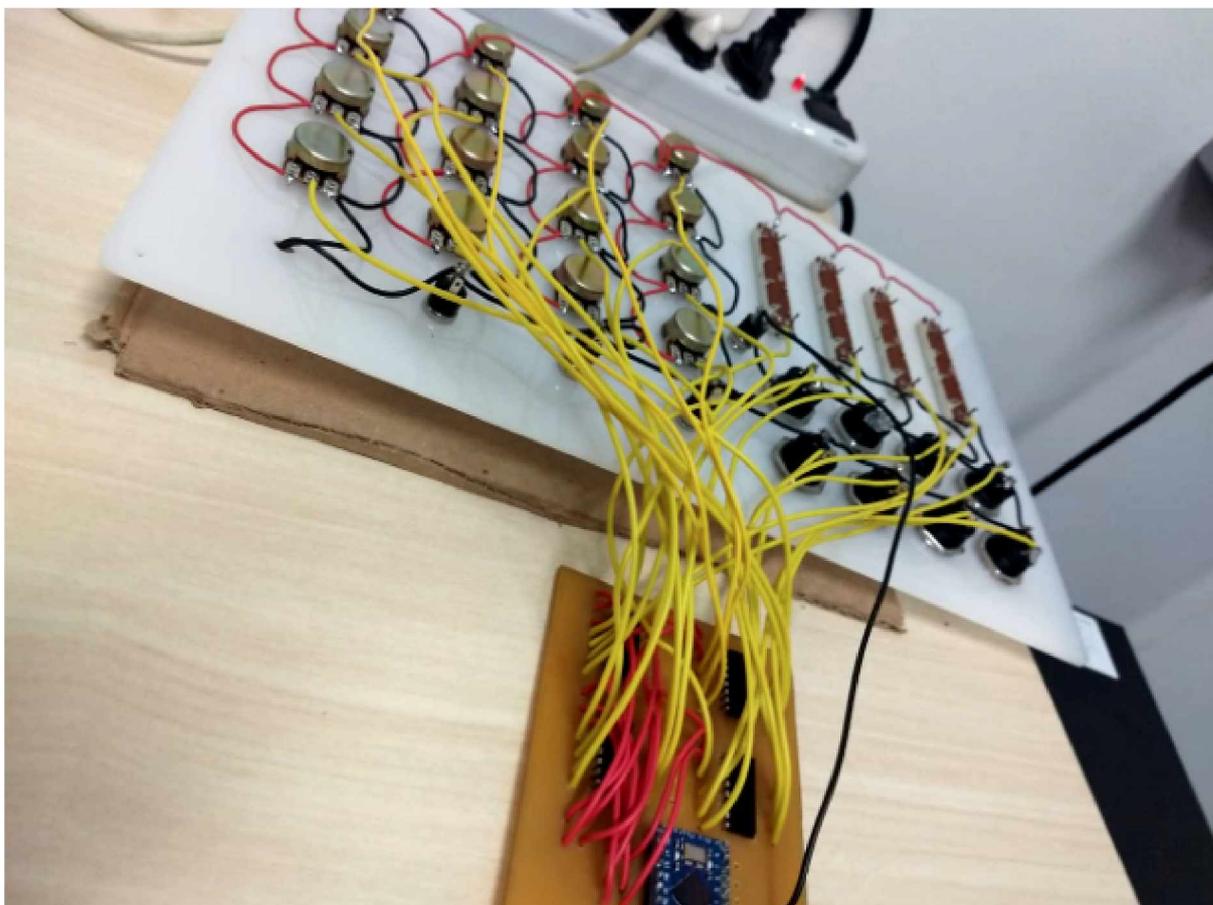


Figura 8 – Montagem dos componentes na chapa de acrílico. Fonte: autoria própria



Figura 9 – Montagem final. Fonte: autoria própria

3.2 Projeto de Software

3.2.1 O protocolo de comunicação

O protocolo utilizado para a comunicação do dispositivo com o software compatível é o MIDI. Portanto, a primeira etapa do desenvolvimento do software do projeto envolveu o estudo das especificações técnicas deste protocolo, disponíveis no manual.

3.2.2 Leitura das entradas

A leitura das entradas dos potenciômetros e botões foi feita utilizando o método de varredura. Um *loop* infinito verifica constantemente o estado de cada entrada e, quando há alguma alteração no estado (por exemplo: um botão foi pressionado ou um potenciômetro movido), é gerada e enviada a mensagem MIDI previamente atribuída a esta entrada.

3.2.2.1 Entradas digitais: botões

Os botões foram ligados ao pino *ground* (0V) e às entradas digitais do Arduino utilizando os resistores de *pullup* internos. Para isso, na função `setup()` os pinos dedicados aos botões foram declarados como `INPUT_PULLUP` da seguinte forma:

```
// Liga o resistor de pullup para os botoes ligados no Arduino
for(int i=0; i<NButtons; i++){
    pinMode(buttonPin[i], INPUT_PULLUP);
}
// Liga o resistor de pullup para o pino de leitura do mux digital
pinMode(READPINMUX1, INPUT_PULLUP);
```

Para a leitura dos botões há apenas dois valores possíveis: nível lógico alto ou baixo. Com esses valores é possível saber quando um botão está sendo pressionado ou não.

Um problema comum na utilização de chaves mecânicas em geral é o efeito *bouncing*, que são as rápidas oscilações entre os níveis lógicos alto e baixo quando há uma mudança de estado. Para evitar problemas com acionamentos indevidos no dispositivo, foi realizado o *debounce* digital. O processo consiste em, assim que detectada uma alteração no estado da chave (quando é pressionada, por exemplo), “atrasar” a leitura desse novo estado em alguns milissegundos. O valor deste “atraso” foi definido empiricamente. Nos testes realizados, 1ms foi suficiente.

O uso de atrasos no código utilizando a função `delay()` do Arduino pode vir a ser um problema, uma vez que o Arduino não suporta processamento paralelo e toda a

execução do software fica parada até que o tempo de atraso definido seja alcançado. Esses atrasos impactariam negativamente na experiência de uso do controlador, pois as funções mapeadas não seriam executadas imediatamente após os controles serem acionados.

Para evitar esse problema, foram utilizados *loops* e variáveis de controle. Após detectada a mudança de estado, a cada iteração do *loop* de leitura o algoritmo verifica se o tempo especificado para o *debounce* já foi atingido. Em caso positivo, é feita a nova leitura e, caso a leitura indique um estado diferente do estado original, é gerada a mensagem MIDI correspondente.

3.2.2.2 Entradas analógicas: potenciômetros

Os potenciômetros foram ligados em multiplexadores, e estes ligados às entradas analógicas do Arduino. A leitura, assim como a dos botões, também foi feita utilizando métodos de varredura.

A diferença em relação aos botões é que as alterações na posição dos potenciômetros resultam em um valor numérico que varia entre 0 e 1023, uma vez que o conversor analógico/digital do Arduino trabalha com uma resolução de 10 bits. As mensagens *control change* do protocolo MIDI utilizam valores de 7 bits, variando entre 0 e 127, portanto foi necessário realizar uma conversão entre essas escalas. A função `map()` do Arduino permite fazer esse tipo de conversão de maneira fácil.

3.2.3 Função “shift”

Para aumentar a quantidade de funções do controlador sem aumentar a quantidade de botões foi criada uma função denominada “shift”. Esta função funciona de maneira semelhante à que é encontrada em teclados de computador.

Ao pressionar uma tecla modificadora o software atribui um valor (nota MIDI) diferente para cada botão, permitindo assim que seja mapeado para controlar outra função do software. A função “shift” é desligada quando a tecla modificadora é acionada novamente.

Para que o usuário saiba quando esta função está ativa ou não, os LEDs foram programados para piscarem durante sua ativação.

3.3 Testes

Para verificar o comportamento do dispositivo e analisar se as saídas geradas por ele estavam de acordo com o planejado foi utilizado o software MIDI Monitor. É uma aplicação gratuita para macOS, que funciona como um *sniffer*, analisando e exibindo na tela toda a comunicação trocada entre dispositivos MIDI e o computador. A Figura

10 mostra a tela do MIDI Monitor com as mensagens *note on* e *note off* enviadas pelo controlador ao pressionar e soltar os botões.

A integração com softwares compatíveis foi realizada em dois dispositivos diferentes: Um MacBook Pro com sistema operacional macOS 10.14.1 Mojave, e um celular Motorola Moto Z2 Play com Android Oreo 8.0. Os testes são descritos a seguir.

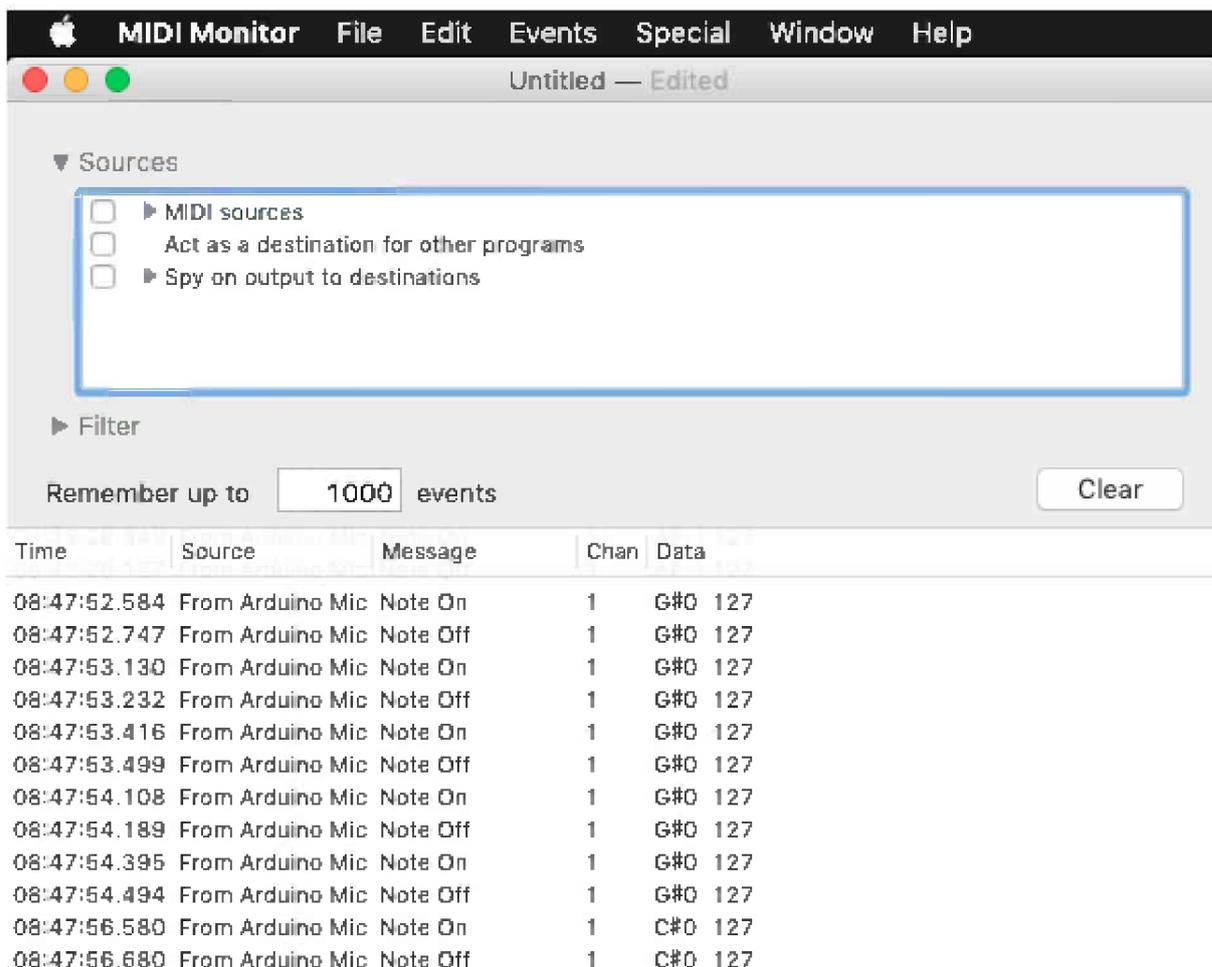


Figura 10 – MIDI Monitor. Fonte: autoria própria

3.4 Integração com software compatível com o protocolo MIDI

3.4.1 Traktor Pro

O Traktor Pro é utilizado por DJs em apresentações ao vivo. É um software comercial com versões para sistemas macOS e Windows. Foi utilizado para os testes no MacBook Pro.

O processo de mapeamento, ou seja, atribuição de um controle MIDI a uma função do software, é feito de forma diferente de acordo com o software que está sendo utilizado. No Traktor, é feito na tela de preferências do software.

A escolha das funções fica a critério do utilizador. Para fins de teste, foram mapeadas as funções mais utilizadas para mixagens ao vivo: controles do *browser* (seleção das músicas para tocar), volume, efeitos, equalizador, filtro, botões de Play, CUE e Loop, indicados na Figura 11.

Os controles do *browser* foram mapeados para os mesmos botões utilizados para os controles de Play e CUE e Loop. Isso foi possível devido a função “shift” do controlador, que permite mapear mais de um controle para um mesmo botão, alternando o seu uso quando necessário.

O teste consistiu em utilizar o controlador para selecionar duas músicas em uma *playlist*, carregar cada uma em um *player*, aplicar e desligar efeitos (alterando os parâmetros de cada um em tempo real) e realizar a mixagem (transição) de uma música para outra.

Adicionalmente, o botão “shift” foi mapeado para alterar o *layout* do software, aumentando a área de visualização do *browser* para facilitar a seleção das músicas, como mostra a Figura 12. O *layout* é restaurado para o original no momento em que a função é desabilitada no controlador.

Todas as funções foram executadas com êxito e sem atrasos entre o acionamento dos controles e a resposta na tela.

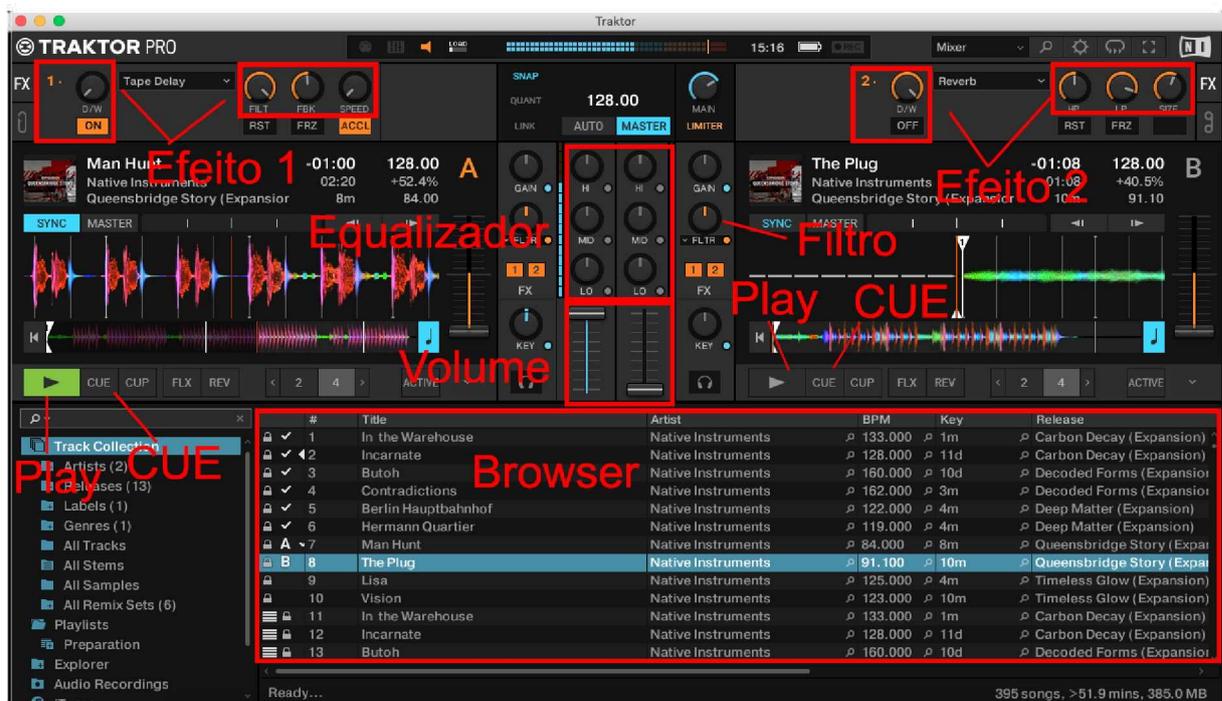


Figura 11 – Tela principal do Traktor Pro 3. Fonte: autoria própria

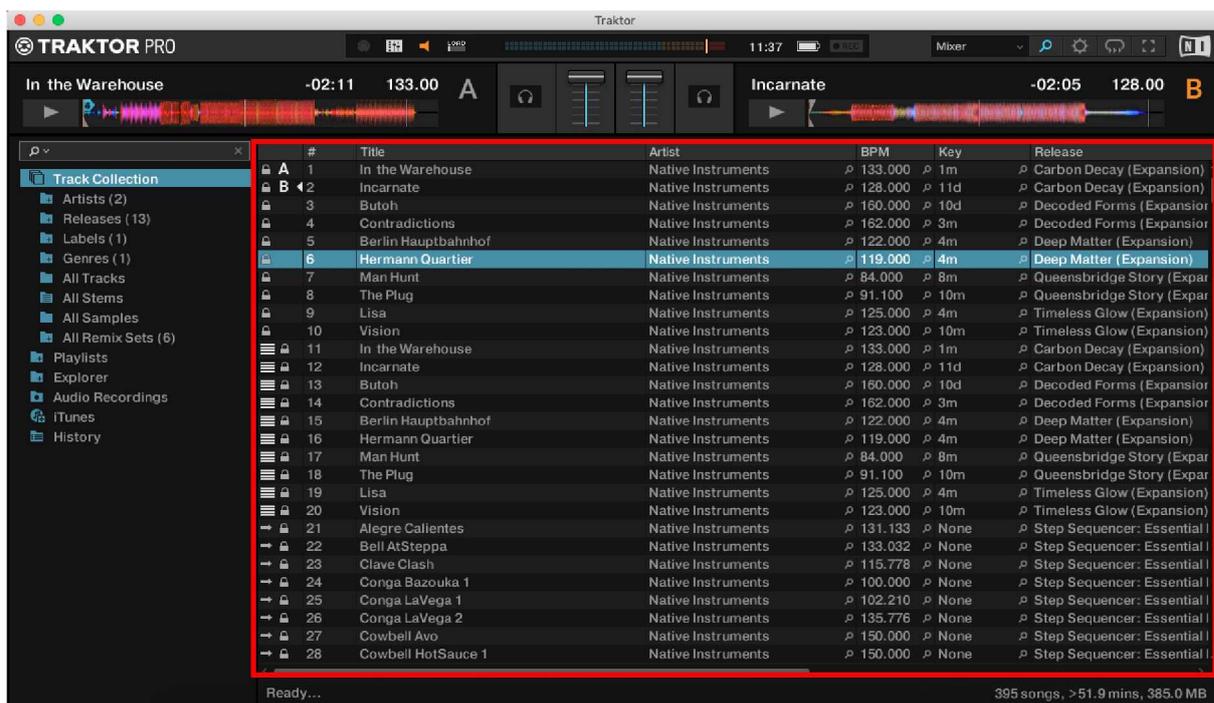


Figura 12 – Traktor Pro com área do *browser* expandida. Fonte: autoria própria

3.4.2 Caustic

O software Caustic é um DAW para sistema operacional Android. A versão gratuita limita a utilização dos controles MIDI a apenas 10 minutos, porém este tempo foi suficiente para realizar os testes e comprovar a compatibilidade do controlador com um outro sistema operacional e outro software que utiliza o protocolo MIDI.

Diferente do Traktor, que é utilizado para tocar e mixar músicas ao vivo, o Caustic é mais utilizado para produzir batidas ou até músicas completas, sequenciando diversos instrumentos em uma espécie de linha do tempo. A Figura 13 mostra a tela de sequenciamento do Caustic.

O teste consistiu em mapear os controles de um dos instrumentos virtuais do Caustic, o Beatbox, mostrado na Figura 14. Durante os 10 minutos de testes, todas as funções mapeadas funcionaram corretamente e sem atraso.

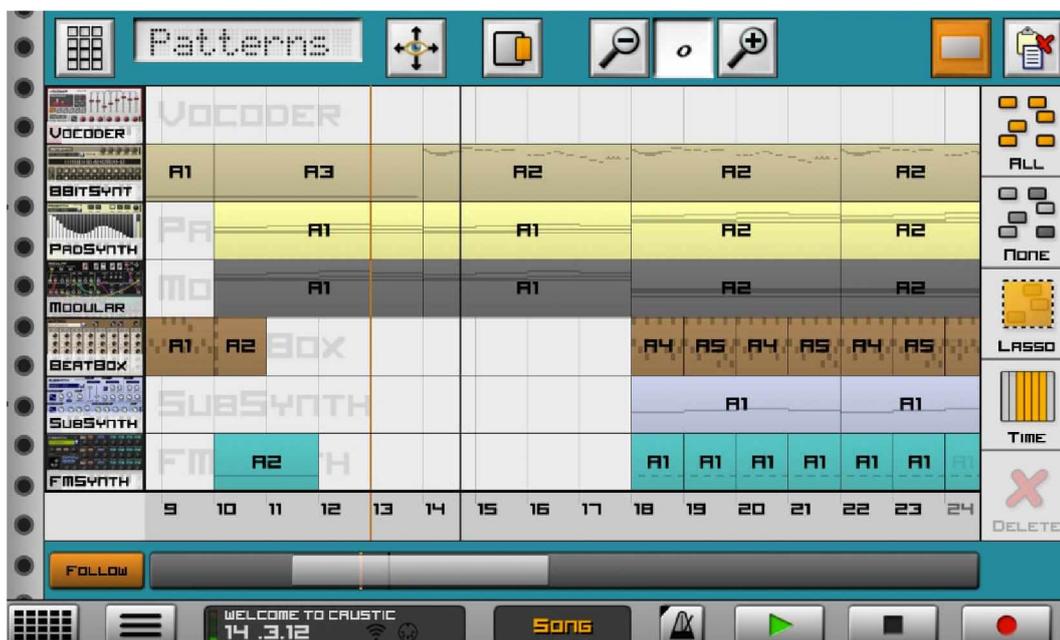


Figura 13 – Caustic 3 - linha do tempo. Fonte: autoria própria



Figura 14 – Caustic 3 - BeatBox. Fonte: autoria própria

4 Considerações Finais

4.1 Custos do projeto

Os custos para a construção do controlador foram:

Componentes Eletrônicos

- 1x Arduino Pro Micro - R\$ 30,00
- 16x Potenciômetro giratório 10k Ω - R\$ 24,00
- 16 Knobs p/ potenciômetro giratório + frete - R\$ 62,00
- 4x Potenciômetros deslizantes 10k Ω - R\$ 26,00
- 8x Botão 19mm - R\$ 20,00
- 5x Botão 8mm - R\$ 25,00
- 4x Multiplexador CD4051B - R\$ - 8,00
- 4x LED alto brilho + resistores - R\$ 4,00
- 1x Placa de fenolite face simples 10x15cm - R\$ 4,50
- 10x Fio 0,25mm (metro) - R\$ 7,50

Componentes Estruturais:

- 1x Chapa de acrílico 3mm + corte a laser - R\$ 30,00
- 1x Chapa de aço inox 430 escovado 1mm + corte a água - R\$ 70,00
- 1x Caixa de madeira personalizada - R\$ 50,00

Valor total: aproximadamente R\$ 361,00.

4.1.1 Comparativo de preços de produtos similares

Foi feita uma cotação online dos produtos comerciais com funções semelhantes às desse projeto, para fins de comparação de preço, como vemos a seguir. A discrepância dos valores à primeira vista se justificaria pelo fato desses produtos comerciais contarem com uma interface de áudio interna para permitir a pré-escuta nos fones de ouvido, função importante nos softwares de discotegem. Entretanto, essa funcionalidade pode ser suprida com o uso de uma interface de áudio USB comprada separadamente por um preço menor.

4.1.1.1 Traktor Kontrol Z1

O Traktor Kontrol Z1, mostrado na Figura 15 é um controlador MIDI fabricado pela Native Instruments. Possui 13 controles giratórios, 3 deslizantes e 5 botões momentâneos. O preço médio encontrado em lojas online brasileiras foi R\$ 1.398,00.

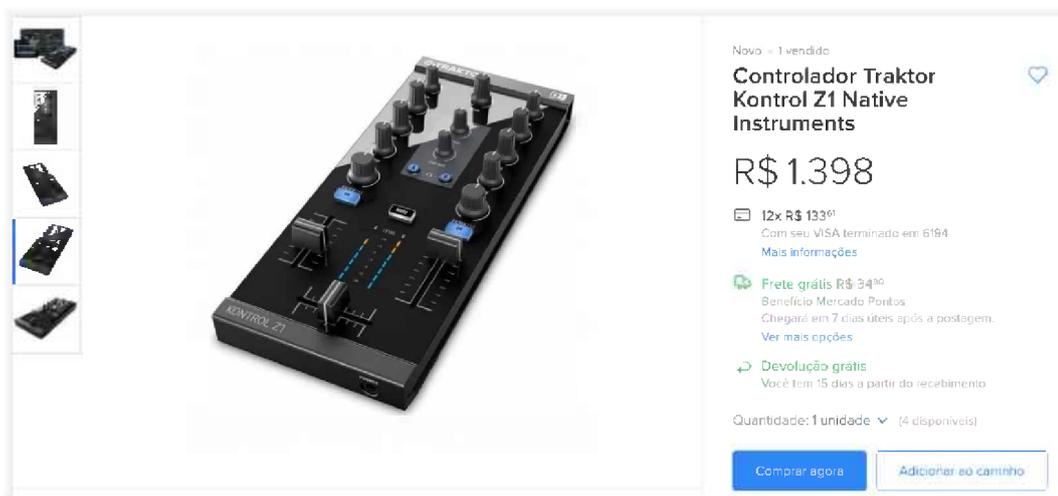


Figura 15 – Anúncio de venda do Traktor Kontrol Z1. Fonte: (MERCADOLIVRE, 2018)

4.1.1.2 Hercules DJ Control Compact

O segundo produto comparado foi o DJ Control Compact, mostrado na Figura 16. Trata-se de um controlador com 6 controles giratórios, 1 deslizante, e 18 botões. O preço médio encontrado em lojas online brasileiras foi R\$ 467,90.

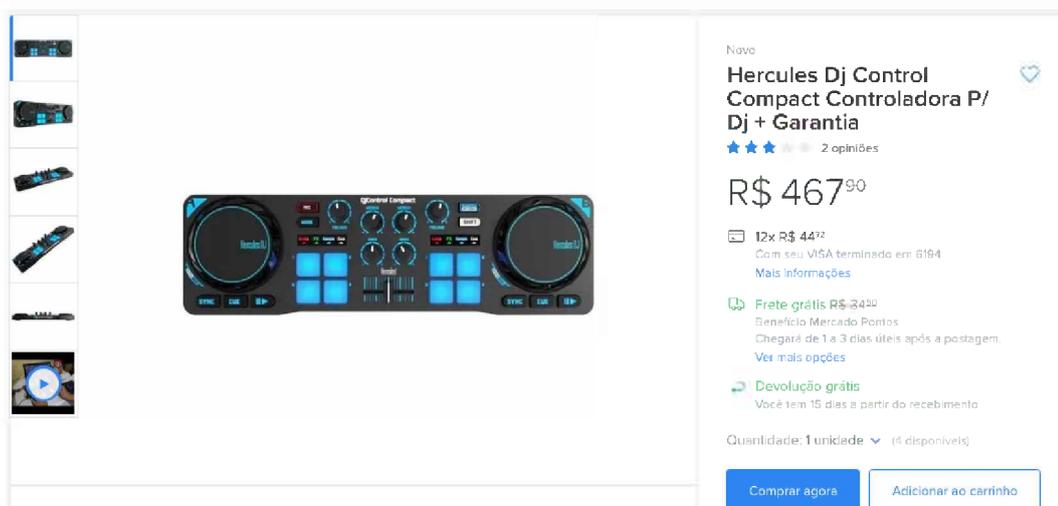


Figura 16 – Anúncio de venda do Hercules DJ Control Compact. Fonte: (MERCADOLIVRE, 2018)

4.2 Dificuldades e Desafios

As dificuldades no desenvolvimento deste projeto foram em sua maioria relacionadas à construção física do dispositivo. A chapa de acrílico e vários componentes eletrônicos foram comprados em outra cidade devido à grande diferença de preço, e outros comprados pela internet devido a dificuldade de se encontrar em lojas físicas.

Após a finalização da montagem, o Arduino utilizado apresentou problemas na conexão micro USB, que por ser muito frágil acabou soltando devido às várias conexões e desconexões do cabo durante os testes. O processo de substituição do Arduino defeituoso foi bastante complexo, uma vez que o componente foi soldado à placa sem possibilidade de remover. O componente precisou ser quebrado para que fosse removido da placa de circuito e substituído. Para evitar este problema futuramente, foi feita uma extensão com um cabo com conexão micro USB, ligado ao Arduino, e uma conexão USB tipo B fêmea, mais resistente.

A confecção da placa de circuito impresso também foi um desafio, pois teve que ser reprojeta para possibilitar sua fabricação em uma placa de face simples e utilizando métodos e ferramentas caseiras.

4.3 Trabalhos Futuros

Como trabalho futuro, seria interessante utilizar *encoders* digitais ao invés de potenciômetros analógicos, pois com isto seria possível mapear mais funções a um único controle físico. Outra melhoria a ser implementada seria a confecção de uma placa de circuito dupla face com componentes SMD (Dispositivo de Montagem Superficial, do inglês *Surface-mount Device*), pois ocupam menos espaço e são soldados diretamente na superfície da placa, sem necessidade de furos ou fios de ligação.

O uso de sensores baseados em proximidade também é uma alternativa pra criar controles que não dependam de toque.

4.4 Conclusão

Após pesquisar uma quantidade considerável de informações sobre os assuntos relacionados ao projeto e então criar o dispositivo, concluiu-se ser possível construir um controlador MIDI personalizado, compatível com diferentes *softwares* e sistemas operacionais, e de custo consideravelmente menor que as opções comerciais disponíveis atualmente.

Mapear as funções de um software em controles físicos aumenta as possibilidades de utilização do mesmo, já que, ao contrário de quando é usado um mouse, mais de uma

função pode ser acionada ao mesmo tempo com as duas mãos.

A disposição dos botões e potenciômetros permitiu uma utilização confortável e ergonômica enquanto a iluminação de LEDs, além de permitir visualizar facilmente os controles em um ambiente escuro, também fornece feedback sobre a função que o controlador está executando no momento.

O tempo de resposta é um ponto crucial para um controlador que será usado em situações ao vivo, por este motivo o software foi programado de forma a reduzir a latência ao mínimo possível. O resultado obtido foi bastante satisfatório, durante os testes as ações realizadas no controlador refletiram instantaneamente nas funções mapeadas no software, com uma latência imperceptível.

O trabalho se encerra de forma que todos os objetivos específicos definidos inicialmente foram atingidos, uma vez que foram utilizados conceitos de eletrônica digital para o projeto, bem como softwares EDA e CAD para criação da placa de circuito impresso e dos cortes automatizados, respectivamente, e também foi programado um microcontrolador, especificamente o Arduino, para que funcione como um dispositivo MIDI USB independente, conforme detalhado no capítulo de desenvolvimento.

Referências

- ANDERTON, C. *Article: A brief history of MIDI*. 2013. Disponível em: <<http://www.harmonycentral.com/articles/a-brief-history-of-midi>>. Acesso em: 12 mai. 2018. Citado na página 9.
- ARDUINO. *Arduino: Introduction*. 2018. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 12 mai. 2018. Citado na página 12.
- ARDUINO. *Arduino Forum*. 2018. Disponível em: <<https://forum.arduino.cc>>. Acesso em: 12 mai. 2018. Citado na página 12.
- Arduino Blog. *Traktorino is an open-source DIY MIDI controller for DJs*. 2018. Disponível em: <<https://blog.arduino.cc/2018/04/03/traktorino-is-an-open-source-diy-midi-controller-for-djs/>>. Acesso em: 24 jun. 2018. Citado 2 vezes nas páginas 5 e 14.
- FLOYD, T. L. *Sistemas digitais [recurso eletrônico]: fundamentos e aplicações*. 9. ed. Porto Alegre: Bookman, 2007. Citado na página 12.
- GRATTON, A. *The Hairless MIDI to Serial Bridge*. 2011. Disponível em: <<http://projectgus.github.io/hairless-midiserial/>>. Acesso em: 24 jun. 2018. Citado na página 14.
- GREWAL, G. *MIDIUSB: A MIDI library over USB, based on PluggableUSB*. 2015. Disponível em: <<https://github.com/arduino-libraries/MIDIUSB>>. Acesso em: 24 jun. 2018. Citado na página 14.
- HIDUINO: Native USB-MIDI on the Arduino. 2011. Disponível em: <<https://github.com/ddiakopoulos/hiduino>>. Acesso em: 24 jun. 2018. Citado na página 14.
- HUBER, D. M. *The MIDI Manual: A Practical Guide to MIDI in the Project Studio*. 3. ed. Burlington: Focal Press, 2007. Citado na página 9.
- MANCINI, O. M. M. *UMA BREVE HISTÓRIA DO MIDI - Tradução adaptada do artigo de Kristopher D. Giesing - Music 154*. 2018. Disponível em: <http://www.iar.unicamp.br/disciplinas/am005_2003/midi.pdf>. Acesso em: 12 mai. 2018. Citado na página 9.
- MERCADOLIVRE. 2018. Disponível em: <<https://www.mercadolivre.com.br/>>. Acesso em: 13 dec. 2018. Citado 2 vezes nas páginas 5 e 31.
- MIDI MANUFACTURERS ASSOCIATION. *The Complete MIDI 1.0 Detailed Specification: Incorporating all recommended practices*. 3. ed. [S.l.], 2014. Disponível em: <<https://www.midi.org/specifications>>. Acesso em: 12 mai. 2018. Citado na página 11.
- SILVEIRA, G. *Nerd Musician*. 2018. Disponível em: <<https://www.musiconerd.com/>>. Acesso em: 24 jun. 2018. Citado na página 14.

TEXAS INSTRUMENTS. *CD405xB CMOS Single 8-Channel Analog Multiplexer/Demultiplexer With Logic-Level Conversion*. [S.l.], 1998. Disponível em: <<http://www.ti.com/lit/ds/symlink/cd4051b.pdf>>. Acesso em: 12 mai. 2018. Citado 2 vezes nas páginas 5 e 13.

THE MIDI ASSOCIATION. *The MIDI Association*. 2018. Disponível em: <<https://www.midi.org>>. Acesso em: 12 mai. 2018. Citado na página 11.