



Universidade Federal de Pernambuco

Centro de Informática

Pedro Augusto Torres de Moraes Coutinho

Luva Controladora MIDI

Trabalho de Graduação

Orientador: Giordano Cabral (grec@cin.ufpe.br)

Recife

2016

Agradecimentos

Agradeço a minha família por ter me dado o apoio que eu precisava durante toda a minha carreira acadêmica. Aos meus professores, que me guiaram e me forneceram com tudo aquilo que eu precisava para estar aqui agora.

Resumo

Controladores MIDI são dispositivos capazes de enviar uma nota musical computadorizada através de diferentes interfaces, neste caso o dispositivo sendo estudado se trata de uma luva vestível com contatos nas pontas de cada dedo, um sistema desse tipo é chamado de Interface de Expressão Musical.

Ao unir as pontas dos dedos o sinal elétrico é enviado para o computador que irá interpretar o sinal como uma nota musical, isto irá depender de como foram programadas as combinações. Um usuário é capaz de usar o controlador para emular a forma como naturalmente tocamos bateria em uma mesa com os nossos dedos. Neste relatório será demonstrada a implementação, diferentes funcionalidades e também formas nas quais o dispositivo pode ser usado para criação de música.

Abstract

MIDI Controllers are devices capable of sending digital musical notes through different interfaces, in this case the device being studied is a wearable glove with contacts on each fingertip, such a system can be classified as an Interface for Musical Expression.

By pushing together the fingertips, an electric signal is sent to the computer which will interpret as a musical note which will depend on how the different combinations were programmed. A user should be able to use the controller as a way to emulate how we naturally play drums while tapping our fingertips against a table. This report will demonstrate the implementation, different functionalities as well as ways in which the device can be used to make music.

Sumário

Agradecimentos.....	1
Resumo	2
Abstract	3
Sumário.....	4
1. Introdução	5
2. Fundamentos	7
2.1. Arduino	7
2.2. O Protocolo MIDI	10
2.3. Instrumentos Musicais Digitais	13
2.4. Tecnologia Vestível	15
3. O Estado da Arte.....	16
4. O Controlador	18
4.1. Implementação	20
Modo Simples	20
Modo Dedo Modificador.....	21
Modo Combinatorial	22
5. Implementação	23
5.1 Software no Arduino	24
5.2 Hardware da Luva e circuitos	26
5.3 Conectando ao computador	27
Conclusão.....	28
Referências	29
Código do Arduino para o Modo Simples	32
Código do Arduino para o Modo Modificador	33

1. Introdução

Formas de expressão musical usando as nossas mãos são evidentes em diversas situações, informais ou não. Registros históricos das mãos sendo usadas como um instrumento musical são antigas, porém uma forma em particular é interessante para o contexto deste trabalho. Se trata da mão guidoniana^[1], uma forma mnemônica de expressar notas musicais usando a mão, criada pelo monge Guido de Arezzo no século XI.

Figura 1 - Mão Guidoniana servia como um guia para músicos.



Fonte: Página da Wikipédia da Mão Guidoniana¹.

¹ Disponível em: <https://upload.wikimedia.org/wikipedia/commons/8/8d/Guidonian_hand.jpg> Acesso em dez. 2015.

A mão guidoniana, em seu tempo, provou ser um método eficaz para memorização de notas^[8], porém atualmente é possível transformar esta interface em um meio interativo no qual o músico não se restringe às notas que ele é capaz de cantar através de tecnologia vestível.

Mais recentemente, no século XX, começamos a descobrir as possibilidades da computação vestível^[2], dispositivos em miniatura que podem ser usados sobre ou junto as roupas que nos últimos anos tem se tornado mais popular devido aos avanços na tecnologia que permitem microcontroladores cada vez menores e precisos^[3].

Computadores vestíveis nos permitem criar um instrumento que torna nossas mãos instrumentos musicais^[8], no caso deste trabalho, uma luva programável capaz de distinguir cada dedo das mãos. Como a maioria dos instrumentos musicais digitais, a luva é capaz de enviar sinais MIDI, neste caso através de um interpretador intermediário no computador.

Os sinais criados pela luva são enviados através de um módulo Arduino, um microcontrolador *open-source* capaz de receber e enviar sinais entre diferentes dispositivos^[4], como por exemplo um sinal elétrico criado pelo circuito de dois dedos entrando em contato. Ao criar este sinal, o programa carregado dentro do microcontrolador é responsável por identificar quais foram os dedos usados e qual é a nota que deverá ser enviada em formato MIDI para o dispositivo de saída de som.

Instrumentos musicais digitais são estudados extensivamente pelo grupo MustIC da Universidade Federal de Pernambuco e também pela conferência internacional New Interfaces for Musical Expression (NIME)^[5], que acontece anualmente para discutir o estado da arte em expressão musical, seja digital ou não.

Ao longo desta introdução serão descritos cada um dos elementos que compõem o instrumento: o protocolo MIDI, a plataforma Arduino, tecnologia vestível e instrumentos musicais digitais.

2. Fundamentos

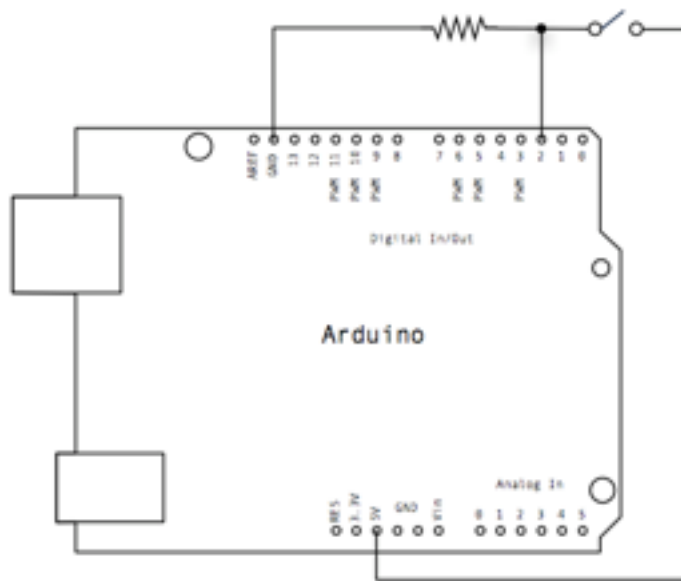
Os seguintes tópicos servem como uma introdução a cada um dos componentes utilizados no trabalho. A seção serve como referência para demonstrar o básico sobre cada tecnologia para que possam ser mostrados maiores detalhes na seção de implementação.

2.1. Arduino

O microcontrolador utilizado no momento é o Arduino UNO R3, apesar de ser uma das unidades mais básicas disponíveis, ele atende todas as necessidades do projeto. A linguagem usada é a linguagem padrão do Arduino, uma mistura de funções de C e C++^[4]. Mais detalhes sobre a implementação do projeto podem ser vistos no capítulo de implementação, por enquanto iremos focar em apresentar a plataforma.

Existem alternativas ao Arduino, porém por causa da sua popularidade e também facilidade de obtenção, ele foi escolhido para este projeto. Existem placas especificamente feitas para computadores vestíveis, como a Lilypad dos mesmos criadores do Arduino. Os princípios mostrados aqui se aplicam da mesma forma, apesar de serem placas diferentes.

Figura 2: Esquema de pinos do Arduino UNO



Fonte: Documentação da plataforma Arduino.³

No esquema da Fig. 4 está sendo usado o GND e a linha de 5V da própria placa para criar um circuito que é lido pelo pino de número 2. Um resistor é colocado para evitar interferência de sinal, por se tratar de uma placa pequena é possível que haja interferência quando muitos pinos estão sendo usados.

A maioria dos programas em Arduino são divididos em três blocos^[15]:

- Declaração de variáveis globais: Informações que são importantes e devem ser guardadas
- `setup()`: Função onde é feita a configuração inicial do programa, como por exemplo a escolha do modo de cada pino (INPUT ou OUTPUT)
 - Exemplo: `pinMode(PIN_NUMBER, INPUT);`
- `loop()`: Laço de execução principal que se repete enquanto a placa estiver ligada. É nesta etapa que podemos fazer a leitura de cada uma das entradas da luva. A execução dessa função é feita muitas vezes por segundo, a sua velocidade depende da complexidade do código contido.

³ Disponível em: <<https://www.arduino.cc/en/Tutorial/Button>> Acesso em dez. 2015.

Com estas peças já pode ser feito um programa básico no Arduino tendo um pouco de conhecimento de como funcionam circuitos. Para auxiliar neste aspecto, é usada uma Breadboard, uma placa de prototipação de circuitos que não requer soldas.

2.2. O Protocolo MIDI

MIDI, abreviação de *Musical Instrument Digital Interface*, se tornou o principal meio de comunicação entre instrumentos digitais e computadores desde a sua concepção em 1983 e mudou a forma na qual músicos podiam fazer música com computadores^[5]. Hoje é fácil ver o impacto da criação do protocolo na quantidade de instrumentos musicais eletrônicos e também através dos instrumentos musicais digitais, como o que está sendo estudado.

Diferentes notas podem ser enviadas pelo controlador alterando o *pitch*, para o propósito deste projeto, iremos usar uma velocidade e um canal fixo, no entanto as notas podem ser alteradas simplesmente modificando o valor em cada função. A implementação é feita através da biblioteca *Arduino MIDI Library*, capaz de traduzir diferentes tipos de entrada em notas. Funciona da seguinte maneira:

```
MIDI.sendNoteOn(42,127,1);  
// Envia através da porta serial uma nota com pitch 42, velocidade 127 no canal 1  
MIDI.sendNoteOff(42,0,1);  
// Envia através da porta serial um sinal para encerrar a nota com pitch 42 no canal 1
```

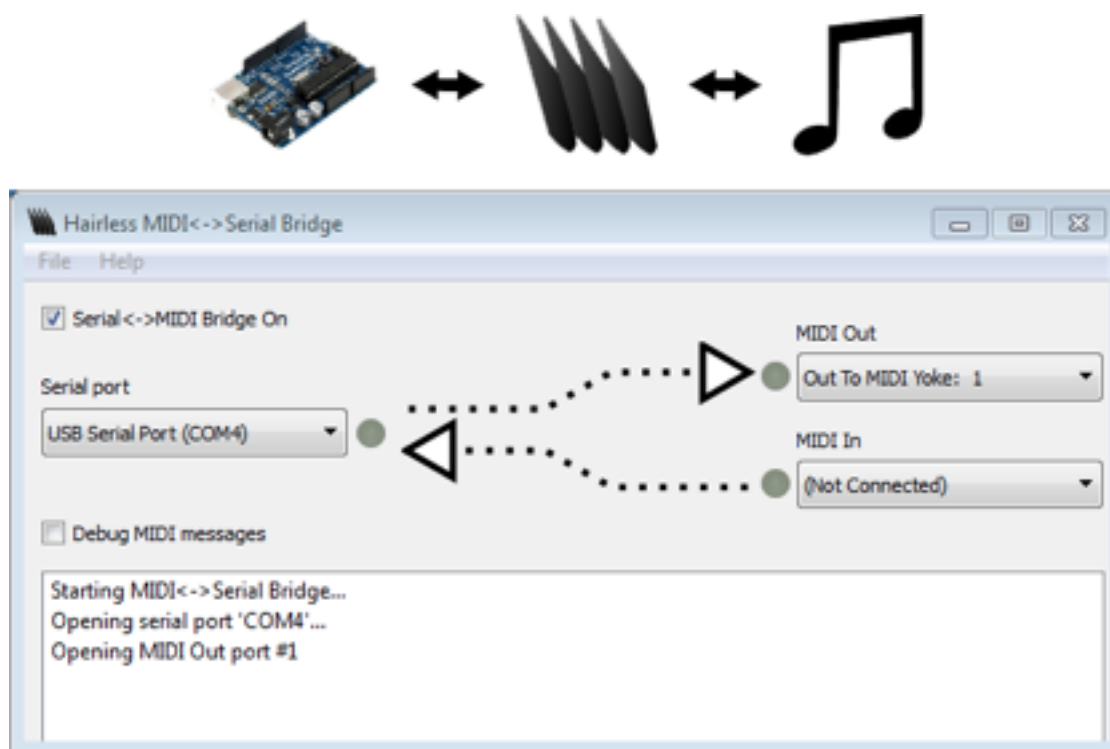
Nem todos os instrumentos necessitam de uma mensagem para indicar que a nota não está mais sendo tocada, porém por motivos de compatibilidade com um leque maior de instrumentos recomenda-se o uso da mensagem Note-OFF^[5].

Porém, apesar de haver a possibilidade de uma comunicação direta do Arduino com um socket MIDI através desta mesma biblioteca, houveram limitações quanto a quantidade de pinos disponíveis no arduino por este projeto. Para ainda ser capaz de

enviar sinais MIDI, foi utilizada a saída serial do próprio microcontrolador. Esta saída serial pode ser transformada em um dispositivo MIDI virtual através de diferentes programas, no caso deste projeto foi escolhido o *Hairless MIDI*, por se tratar de um programa de código aberto feito com o Arduino em mente.

O programa lê a saída serial do Arduino, transforma em um dispositivo MIDI virtual e dá a opção de selecionar um outro dispositivo de saída virtual. Possibilitando o uso da luva em qualquer tipo de programa capaz de interpretar o protocolo.

Figura 4: Demonstração de como funciona o programa Hairless MIDI que faz a comunicação entre a saída serial do Arduino e entradas virtuais MIDI.



Fonte: Página oficial do projeto Hairless MIDI < - > Serial Bridge.⁴

No dispositivo de saída selecionado é onde o sinal MIDI é interpretado. É importante lembrar que o protocolo não envia dados musicais como vistos em arquivos Waveform ou MP3. Neste caso o protocolo está enviando mensagens binárias que

⁴ Disponível em: <<https://projectgus.github.io/hairless-midiserial/>> Acesso em dez. 2015.

devem ser interpretadas de acordo com as informações inseridas em cada função implementada no nosso código.

```
MIDI.sendNoteOn(42,127,1);
```

Quando traduzimos isso para uma mensagem do protocolo MIDI^[16]:

```
10010000, 00101010, 01111111
```

Respectivamente cada um dos bytes:

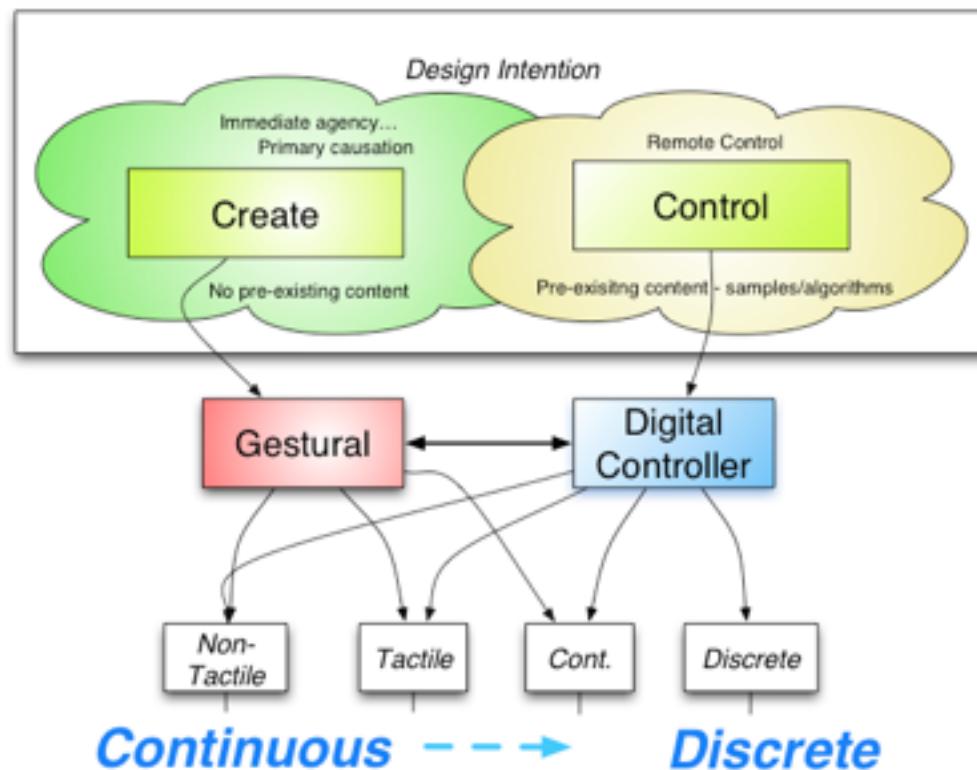
```
Chan 1 Note on, pitch 42, velocity 127
```

2.3. Instrumentos Musicais Digitais

Instrumentos Musicais Digitais, ou a sua abreviação em inglês DMI, abrangem uma gama de diferentes formas de interagir com a música através de computadores.

Existem diferentes atributos que definem a forma na qual um DMI pode se encaixar, a figura abaixo ilustra como esses instrumentos podem funcionar:

Figura 5: Diagrama de classificação e descrição de DMI's



Fonte: Página da University of Western Sydney dedicada ao estudo de DMI's⁵

⁵ Disponível em: <<http://vipre.uws.edu.au/dmi>> Acesso em dez. 2015.

Como demonstrado na *Fig 5*, existem interseções entre as diferentes características, porém levando este esquema em mente, podemos entender quais são as principais características de um instrumento musical digital.

- Intenções de design:
 - Criação
 - Onde o conteúdo gerado pelo instrumento é criado pelo músico, da mesma forma que notas são tocadas em um piano, também podem ser tocadas através de mensagens MIDI em uma interface desse tipo.
 - Controle
 - Onde o músico age como um controlador de amostras já existentes, instrumentos como Samplers digitais ou interfaces que aplicam algoritmos para modificar uma música sendo tocada.
- Interfaces Gestuais
 - Estas não somente se manifestam como interfaces de detecção de gestos através de ferramentas de gestos virtuais como Kinect ou LEAP Motion, mas também podem ser gestos detectados através de um meio físico. Uma luva como controlador MIDI pode ser considerada uma interface de gestos^[8].
- Interfaces de controle digital
 - Interfaces que funcionam através de um controle digital, exemplos podem incluir instrumentos que usam botões ou telas sensíveis ao toque.
- Táctil, Não-táctil
 - Se é ou não uma interface física, manuseável.
- Contínuo ou Discreto
 - Da mesma forma como podemos tratar números em intervalos contínuos ou discretos, também podemos analisar o espectro musical da mesma forma.

Como mostra o Diagrama, essas características não são mutuamente exclusivas, um DMI pode implementar todos estes atributos, dependendo da forma na qual ele é programado. Instrumentos digitais podem ser tácteis e não-tácteis ao mesmo tempo se ele incorpora diferentes tipos de interação^[20].

Um instrumento digital também não necessariamente incorpora o protocolo MIDI, pois existem diferentes maneiras de mapeamento de interfaces em música. Um maestro digital, por exemplo, pode se comunicar enviando sinais de mudança de frequência através de um protocolo próprio ou mudança de parâmetros em um algoritmo.

2.4. Tecnologia Vestível

Seguindo uma tendência histórica de criar computadores cada vez menores e discretos, em 1955 foi registrado um dos pioneiros em vestimentas inteligentes. Edward Thorp criou um computador que em 1961 era capaz de aumentar suas chances de ganhar no blackjack em 44% e que cabia dentro do sapato como uma palmilha^[2].

Desde então computadores vestíveis têm assumido diferentes formas, um progresso proporcionado principalmente pela popularidade dos microcontroladores e diferentes tipos de tecidos condutivos^[17].

No âmbito de instrumentos musicais digitais a computação vestível é um assunto pertinente, com seu primeiro artigo aparecendo em 2003^[18]. Desde então tem sido recorrentes as demonstrações de computadores vestíveis, principalmente por proporcionarem uma forma de expressão mais natural de interação com a música, diminuindo as barreiras entre o artista e a música da mesma forma que a mão guidoniana tinha a proposta de usar a mão como um instrumento^[9].

No entanto, um dos maiores desafios está no mapeamento de ações em uma interface vestível^[8], principalmente ao considerar as diferentes limitações de sensores e as intenções de cada usuário. Este cenário faz com que grande parte das novas interfaces de expressão musical se tornem uma ferramenta principalmente usada pelo seu próprio criador^[19].

3. O Estado da Arte

Nesta seção será elaborado o atual estado da arte no que se trata de DMI's. Também serão mencionadas diferentes tipos de interfaces de expressão musical por seu relacionamento com o assunto.

Como diferentes ferramentas que operam sobre computadores, instrumentos digitais também se beneficiam do lançamento de novas tecnologias. Microcontroladores menores aumentam a portabilidade dos instrumentos, novas interfaces permitem novas formas de interagir com a música e computadores mais rápidos diminuem o atraso e permitem uma experiência mais fluida^[21].

Anualmente são reunidos os trabalhos do NIME (*New Interfaces for Musical Expression*; Novas Interfaces para Expressão Musical) e demonstrados ao público. Este congresso pode ser visto como o estado da arte para o assunto deste trabalho, pelo fato de explorar diferentes tecnologias de uma forma que possam se tornar instrumentos musicais digitais. Para isto são diferentes plataformas utilizadas, desde Arduino até o Wii. A utilização de ferramentas de detecção de gestos tem sido prevalente por permitir a utilização de gestos familiares aos músicos, porém permitindo diferentes conveniências, no caso do controle do Wii, ele pode ser usado para simular uma alfaia^[21].

Para a implementação desta luva, algumas tecnologias poderiam permitir uma experiência similar ao que foi implementado usando o Arduino, mas cada uma contava

com limitações que as tornaram inviáveis. A seguir serão discutidas as diferentes alternativas à implementação com o Arduino.

LEAP Motion, se trata de um dispositivo de detecção de movimentos para as mãos. A API disponibilizada pelo LEAP Motion permite a detecção de gestos com precisão e latência relativamente baixa para o que ele propõe. Porém existem diferentes barreiras de entrada para trabalhar com a ferramenta, principalmente o aprendizado sobre a API de rastreamento que é usada. Iriam ser necessárias muito mais horas para implementar o mesmo projeto e apesar de ter uma baixa latência quando funcionando no modo de baixa precisão, o Arduino apresenta uma latência muito menor por se tratar de um sistema mais simples que não necessariamente detecta gestos.

Telas multi-toque também permitem um sistema parecido com a luva, porém não permitem a detecção de dedos individualmente. Ao tocar na tela não há uma forma de saber qual dedo foi usado, a não ser que sejam delimitadas áreas para cada extremidade, porém isso iria limitar o projeto ao torná-lo um piano em uma interface de toque. Além disto, também iria limitar o funcionamento do instrumento, por não permitir o uso do polegar como o ponto de toque.

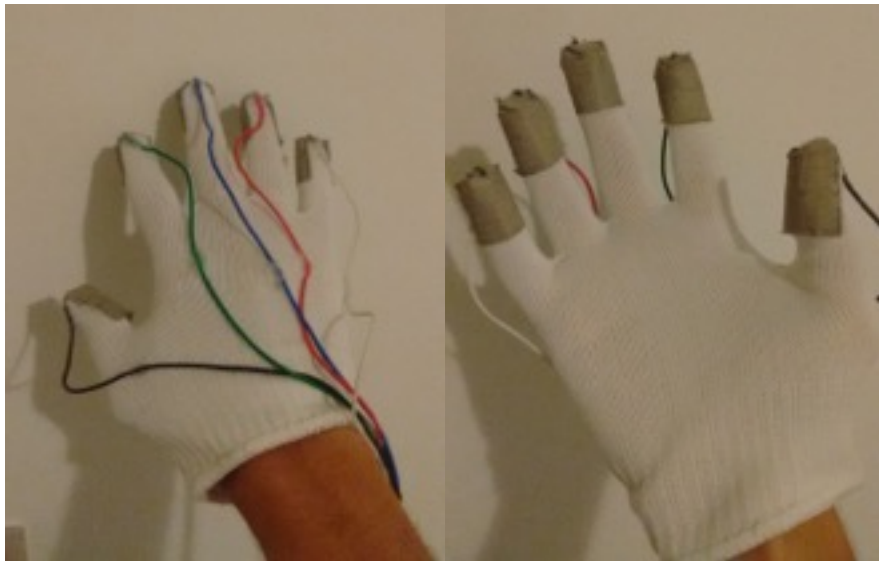
Futuramente a Google pretende lançar no mercado um novo chip de detecção de movimentos de alta precisão, atualmente chamado de **Project Soli**. Este foi apresentado no Google I/O de 2015 e apresenta um grande potencial para instrumentos digitais, porém resta esperar para saber se será uma ferramenta viável para a criação de interfaces^[22].

A conclusão que pode ser tirada dos pontos acima é que apesar de existirem plataformas que atendam todos os requisitos do projeto, o Arduino é potencialmente a melhor escolha por permitir facilidade de implementação quando comparado ao LEAP Motion e flexibilidade quando comparado a telas de toque.

4. O Controlador

O controlador teve como idéia inicial o uso da mão como um instrumento musical sem nenhum intermediário. A forma de produzir notas seria através do contato entre os dedos. Usando o polegar como um ponto de contato, qualquer outro dedo que o toque irá disparar uma mensagem MIDI indicando que houve um sinal. Isto pode ser usado para as mesmas funcionalidades que o protocolo MIDI permite quando conectado a um programa.

Figura 6: O controlador vestido em uma mão.



Fonte: Elaborada pelo autor

O sistema de entrada funciona através de inputs pull-down do Arduino onde cada um dos dedos está sempre com uma corrente de 5V circulando e ao entrar em contato com um terminal GND (Terra) o sistema detecta atividade. As pontas dos dedos são feitas a partir de um tecido condutivo de baixa resistência, permitindo latência muito baixa e alta precisão nos contatos.

Sendo uma plataforma programável onde o usuário pode escolher qual é o comportamento que cada combinação deve ter, isto se torna um instrumento que se

adapta às necessidades do músico. Sejam amostras, notas, comandos ou algoritmos, o usuário pode usar plataformas intermediárias que implementam o protocolo MIDI para criar músicas.

Figura 7: Interface de customização da luva

Diagrama	Pitch
	0
	0
	0
	0
	0

[Exportar código](#)

```

#include <MIDI.h>

//thumb, index, middle, ring, pinky respectively in each array
int pins[] = {13, 2, 4, 8, 11};
int values[] = {0, 0, 0, 0, 0};
int noteStatuses[] = {0, 0, 0, 0, 0};
int pitches[] = {0, 0, 0, 0, 0};

MIDI_CREATE_DEFAULT_INSTANCE();

void setup() {
  for (int i = 0; i < 5; i++) {
    pinMode(pins[i], INPUT);
  }

  MIDI.begin();
  Serial.begin(115200);
}

void loop() {
  for (int i = 0; i < 5; i++) {
    values[i] = digitalRead(pins[i]);
  }

```

OK

Fonte: Elaborada pelo autor

Para uma programação básica do dispositivo, foi criada uma interface onde o usuário pode escolher qual será o pitch de cada comando. Na figura 7 está sendo demonstrada a luva no modo de operação mais simples, onde cada dedo corresponde a uma nota. Do lado esquerdo de cada campo de texto há um diagrama indicando qual é o dedo que deve ser usado para reproduzir a nota correspondente no campo de texto.

Ao apertar em "Exportar código" é demonstrado o programa que deve ser compilado para o Arduino. Ao compilar, o próximo passo é vestir a luva e conectar o Arduino através do Hairless MIDI em um Output como demonstrado na figura 4. Para este projeto, foi utilizado o Ableton Live 9 como saída. No Ableton, assim como em outros programas, uma mensagem MIDI pode ser transformada em uma rotina musical, um algoritmo ou uma amostra, permitindo que esta parte do instrumento não precise ser implementada diretamente no código do microcontrolador.

4.1. Implementação

O instrumento tem três modos de funcionamento implementados. Cada modo foi criado a partir de observações sobre usuários que experimentavam e tentavam interagir de uma forma na qual não foi prevista neste trabalho.

Modo Simples

Este modo é o mais básico onde cada dedo está associado a uma ação. Ao entrar em contato com uma superfície aterrada, a mensagem MIDI é enviada para o programa associado ao controlador.

A idéia inicial da implementação desse modo surgiu como uma interpretação da forma na qual as pessoas usam as mãos para tocar percussão em uma mesa ou qualquer outra superfície.

Por apenas possibilitar uma nota associada a cada dedo, não existe muita viabilidade de uso além de situações mais simples ou instrumentos que não tem uma variedade muito grande de notas, como é o caso de uma bateria que pode ter somente 5 componentes.

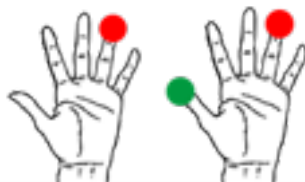
Por ser a forma mais intuitiva de uso, também foi a forma mais fácil de avaliar com usuários inexperientes, que ao vestirem a luva demonstraram imediatamente a intenção na qual foi feito este modo de operação. Também há de se levar em consideração o fato de ser o modo de operação mais simples, portanto o mais rápido de se acostumar e aprender.

Além dessa aplicação em percussão, também há a possibilidade de ser usado como um controlador MIDI de amostras musicais, isto é, trechos pré-selecionados de outras músicas para serem tocados em uma performance musical. Atualmente isto é feito com *Samplers*, instrumentos que se assemelham a um teclado de computador por ter botões na sua face.

Modo Dedo Modificador

Esta forma já apresenta uma complexidade um pouco maior. Neste modo de operação, apenas quatro dedos estão associados a notas, enquanto um deles é usado como um modificador da escala, ou seja, quando o modificador está ativo os outros quatro dedos são associados à mensagens diferentes.

Figura 8: Duas ações associadas ao dedo anelar, a segunda usando o modificador no polegar



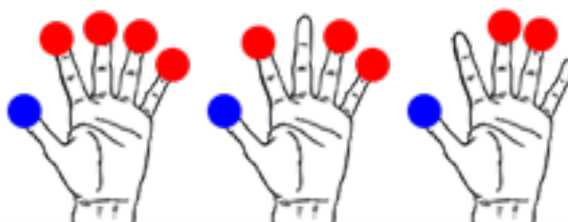
Fonte: Elaborada pelo autor

Na figura 8 é possível ver uma demonstração de quais dedos estão ativos em duas situações diferentes. Comparado ao modo simples, existe uma limitação de uso aparente, que é a impossibilidade de tocar duas notas ao mesmo tempo se elas não estiverem associadas ao mesmo estado do dedo modificador, o que cria uma necessidade de planejamento prévio. Para usuários que não tem prática com o instrumento, este modo já pode ser um desafio.

Modo Combinatorial

De uma forma parecida com o modo anterior, também conta com apenas quatro dedos capazes de enviar mensagens MIDI, porém ao invés de usar o polegar como um dedo modificador ele é usado como o ponto de contato. Isto significa que qualquer outro dedo que entrar em contato com o polegar irá reproduzir uma nota. Além disso, cada uma das possíveis combinações dos quatro dedos é capaz de enviar uma informação diferente, aumentando o número de notas possíveis que eram oito, para quinze.

Figura 9: Três possíveis combinações deste modo de operação



Fonte: Elaborada pelo autor

Durante a fase de experimentação com este modo, foi vista uma dificuldade em realizar ações muito rápidas, não só por causa da quantidade de possíveis combinações mas também pela complexidade. É evidente que para usar o instrumento dessa maneira deve haver prática e também planejamento assim como o modo de dedo modificador.

5. Implementação

A seguir serão demonstrados detalhes da implementação do projeto, desde a sua parte de software até o hardware físico e sua integração com o Arduino. Isto será demonstrado através de trechos de código e figuras dos circuitos.

5.1 Software no Arduino

Assim como outras linguagens, ao trabalhar no Arduino podemos seguir os mesmos padrões de desenvolvimento, isto é, iniciamos o código declarando as variáveis globais:

```
int pins[] = {13, 2, 4, 8, 11};
```

Este *array* guarda informações de quais pinos estão sendo usados para cada dedo. A ordem de cada dedo relacionado aos seus respectivos pinos é: Polegar, indicador, médio, anelar, mindinho. Todos os *arrays* de inicialização são relativos a ordem deste *array*.

```
int values[] = {1, 1, 1, 1, 1};
```

Aqui são guardados os valores de cada pino do Arduino, caso ele esteja ativo o seu valor muda para 0. O valor é contra-intuitivo por se tratar de um sistema de pull-down, onde o pino está sempre ativo, exceto quando existe um contato com um terminal terra (GND).

```
int noteStatuses[] = {0, 0, 0, 0, 0};
```

Para ser possível manter o estado de cada nota e saber quando é necessário enviar o sinal de On ou Off para cada nota MIDI é usado um array de valores. Caso a nota esteja ativa, o valor correspondente muda para 1.

```
int pitches[] = {$1, $2, $3, $4, $5};
```

Os valores deste *array* são mantidos com um cifrão prefixado para serem modificados através de expressões regulares quando o músico escolher quais serão os pitches para cada dedo.

Com esses *arrays* já é possível recordar o estado do programa para cada ciclo do laço de repetição do Arduino. Para enviar o sinal MIDI de On para cada nota são feitas verificações dos estados de cada variável:

```
if (values[i] == 0 && noteStatuses[i] == 0 && pitches[i] != 0)  
MIDI.sendNoteOn(pitches[i], 127, 1);  
noteStatuses[i] = 1;
```

Se o dedo correspondente ao pino *i* estiver ativo, ou seja, se o seu valor correspondente for 0 e também se a nota não estiver sendo toca no momento. O pitch também é verificado porque se o seu valor for 0 significa que o usuário decidiu não associar uma nota a ação.

Para o sinal de Off, uma verificação similar é feita:

```
else if (values[i] == 1 && noteStatuses[i] == 1)  
MIDI.sendNoteOff(pitches[i], 0, 1);  
noteStatuses[i] = 0;
```

Caso o valor do pino seja 1 (que quer dizer inativo) e também se a nota estiver ativa, pois o envio de mensagens Off para notas que não estão sendo tocadas irá atrasar a execução do programa.

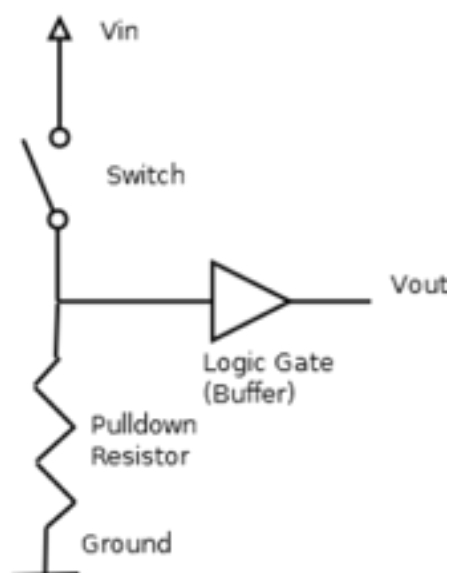
Usando esta lógica, é possível criar diferentes modos de operação para o programa sem modificações muito grandes. O código completo pode ser visto no apêndice A.

5.2 Hardware da Luva e circuitos

Para conduzir o sinal elétrico através da luva foi necessário usar um componente condutivo nas pontas dos dedos. Para este trabalho foi escolhido uma malha de prata para embrulhar cada dedo e conduzir a corrente da placa através de um fio condutor.

No Arduino os botões precisam de resistores para garantir que as entradas estarão funcionando com uma certa impedância de forma que não haja interferência entre os contatos. Para cada pino de entrada que é usado, também é necessário um resistor, porque o simples fato de não haver nada conectado a uma entrada não irá garantir que ela está mostrando o valor correto^[23].

Figura 10: Esquema do circuito dos botões no Arduino.



Fonte: Pull Up and Pull Down Resistors - Arduino Learning⁶

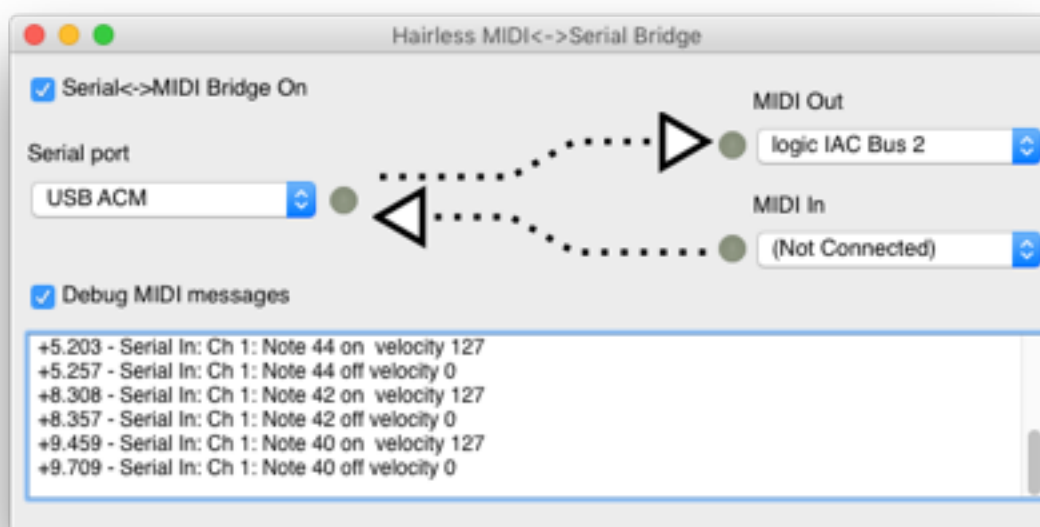
No momento que todos os circuitos estão conectados resta apenas ligá-los ao microcontrolador nos pinos desejados.

⁶ Disponível em: <<http://playground.arduino.cc/CommonTopics/PullUpDownResistor>> Acesso em dez. 2015.

5.3 Conectando ao computador

Com o Arduino conectado via USB já é possível usá-lo como um dispositivo MIDI através de um software intermediário, o Hairless MIDI, que irá transformar a saída serial do Arduino criada pelo método *Serial.begin(115200);* e *MIDI.begin();*. O programa irá interpretar os dados binários em mensagens MIDI. Para se certificar que as mensagens enviadas estão corretas basta marcar a checkbox de debug:

Figura 11: Mensagens de Debug no Hairless MIDI



Fonte: Elaborada pelo autor

Conclusão

O foco deste projeto, de construir um instrumento musical digital vestível, foi atingido. Além disto, também foram estudadas algumas das possíveis formas de uso do instrumento.

Desde a concepção da idéia e a elaboração da proposta do projeto o escopo e os requisitos foram modificados por causa de diferentes modos de operação que poderiam ser incorporados para expandir a forma na qual o instrumento poderia ser utilizado.

Apesar de não ter sido avaliado por músicos profissionais, o seu papel como uma interface de expressão musical foi desempenhado e ainda existe espaço para maiores modificações e melhorias do instrumento para então haver uma validação do seu desempenho como uma ferramenta que realmente possa ser utilizada. Além disso a avaliação de um instrumento musical é um processo que depende da forma na qual os músicos irão utilizá-lo. Por se tratar de um instrumento que pode assumir diferentes formas, o processo de avaliação também é mais demorado.

Os próximos passos de desenvolvimento do instrumento seriam a elaboração de uma luva mais robusta e uma interface que permita ao músico desenvolver uma performance com o instrumento em mente. Além de diminuir o circuito para uma placa apropriada, permitindo uma liberdade maior de movimentação e também proporcionando um instrumento mais robusto.

Referências

- [1] BERGER, Karol. The Guidonian Hand. M. Carruthers and JM Ziolkowski, The Medieval Craft of Memory: An Anthology of Texts and Pictures, 2004, 70.
- [2] THORP, Edward. The Invention of the First Wearable Computer. Iswc '98 Proceedings Of The 2nd Ieee International Symposium On Wearable Computers. Washington, Dc, p. 4-8. out. 1998.
- [3] THOMAS, Bruce. Have We Achieved the Ultimate Wearable Computer? Iswc '12 Proceedings Of The 2012 16th Annual International Symposium On Wearable Computers (iswc). Washington, Dc, p. 104-107. jun. 2012.
- [4] GIBB, Alicia M. New media art, design, and the Arduino microcontroller: A malleable tool. 2010. PhD Thesis. Pratt Institute.
- [5] BARBOSA, Jeronimo, et al. Designing DMIs for Popular Music in the Brazilian Northeast: Lessons Learned.
- [6] BOOM, Michael. Music through MIDI: using MIDI to create your own music system. Redmond, Wa: Microsoft Press, 1988.
- [7] BOVERMANN, Till; HINRICHSSEN, Amelie; LOPES, Dominik Hildebrand Marques. 3DMIN — Challenges and Interventions in Design, Development and Dissemination of New Musical Instruments. Berlin, out. 2014.
- [8] MYLLYKOSKI, Mikko, et al. Prototyping hand-based wearable music education technology.
- [9] TORRE, Giuseppe. The design of a new musical glove: a live performance approach. 2013.

- [10] BARBOSA, Jeronimo, et al. What does "Evaluation" mean for the NIME community?. In: *NIME 2015-15th International Conference on New Interfaces for Musical Expression*. Louisiana State University, 2015. p. 6.
- [11] MIRANDA, Eduardo Reck; WANDERLEY, Marcelo M. *New digital musical instruments: control and interaction beyond the keyboard*. AR Editions, Inc., 2006.
- [12] JORDA, Sergi. Digital Lutherie: Crafting musical computers for new musics performance and improvisation. *PhD diss., Universitat Pompeu Fabra, Departament de Tecnologia*, 2005.
- [13] O'MODHRAIN, Sile. A framework for the evaluation of digital musical instruments. *Computer Music Journal*, 2011, 35.1: 28-42.
- [14] STOWELL, Dan; PLUMBLEY, Mark D.; BRYAN-KINNS, Nick. Discourse analysis evaluation method for expressive musical interfaces. In: *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*. 2008. p. 81-86.
- [15] ARDUINO. Arduino - Tutorials. Disponível em: <https://www.arduino.cc/en/Tutorial/HomePage>. Acesso em dez. 2015.
- [16] MIDI MANUFACTURERS ASSOCIATION. MIDI Message Table 1. Disponível em: <http://www.midi.org/techspecs/midimessages.php>. Acesso em dez. 2015
- [17] BUECHLEY, Leah; EISENBERG, Michael. The LilyPad Arduino: Toward wearable engineering for everyone. *Pervasive Computing, IEEE*, 2008, 7.2: 12-15.
- [18] RYAN, Joel; SALTER, Christopher. TGarden: wearable instruments and augmented physicality. In: *Proceedings of the 2003 conference on New interfaces for musical expression*. National University of Singapore, 2003. p. 87-90.

[19] MEDEIROS, Rodrigo, et al. Challenges in Designing New Interfaces for Musical Expression. In: *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience*. Springer International Publishing, 2014. p. 643-652.

[20] WU, J. Cecilia, et al. Tibetan Singing Prayer Wheel: A Hybrid Musical-Spiritual Instrument Using Gestural Control.

[21] BARBOSA, Jeronimo. Um Framework para avaliação da experiência de uso de instrumentos musicais digitais. Universidade Federal de Pernambuco, 2013.

[22] GOOGLE. Google I/O 2015 - A little badass. Beautiful. Tech and human. Work and love. ATAP. Disponível em: https://www.youtube.com/watch?v=mpbWQbkl8_g. Acesso em dez. 2015.

[23] ARDUINO. :: Pull Up and Pull Down Resistors ::. Disponível em: <http://playground.arduino.cc/CommonTopics/PullUpDownResistor>. Acesso em dez. 2015.

Código do Arduino para o Modo Simples

```
#include <MIDI.h>

//thumb, index, middle, ring, pinky respectively in each array
int pins[] = {13, 2, 4, 8, 11};
int values[] = {0, 0, 0, 0, 0};
int noteStatuses[] = {0, 0, 0, 0, 0};
int pitches[] = {36, 38, 42, 57, 46};

MIDI_CREATE_DEFAULT_INSTANCE();

void setup() {
  for (int i = 0; i < 5; i++) {
    pinMode(pins[i], INPUT);
  }

  MIDI.begin();
  Serial.begin(115200);
}

void loop() {

  for (int i = 0; i < 5; i++) {
    values[i] = digitalRead(pins[i]);

    if (values[i] == 0 && noteStatuses[i] == 0 && pitches[i] != 0) {
      MIDI.sendNoteOn(pitches[i], 127, 1);
      noteStatuses[i] = 1;
    } else if (values[i] == 1 && noteStatuses[i] == 1) {
      MIDI.sendNoteOff(pitches[i], 0, 1);
      noteStatuses[i] = 0;
    }
  }

  delay(10);
}
```

Código do Arduino para o Modo Modificador

```
#include <MIDI.h>

//thumb, index, middle, ring, pinky respectively in each array
const int pins[] = {13, 2, 4, 8, 11};
int values[] = {0, 0, 0, 0, 0};
int noteStatuses[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
const int pitches[] = {$0, $1, $2, $3, $4, $5, $6, $7, $8, $9 };

//defines which key is going to be the modkey
//must be the index in the array
//for instance: 0 for the thumb
int modFinger = 0;

MIDI_CREATE_DEFAULT_INSTANCE();

void setup() {
  for (int i = 0; i < 5; i++) {
    pinMode(pins[i], INPUT);
  }

  MIDI.begin();
  Serial.begin(115200);
}

void loop() {

  for (int i = 1; i < 5; i++) {
    values[i] = digitalRead(pins[i]);

    if (values[i] == 0) {
      values[modFinger] = digitalRead(pins[modFinger]);
      if(noteStatuses[i] == 0 && values[modFinger] == 1 && pitches[i] != 0){
        MIDI.sendNoteOn(pitches[i], 127, 1);
        noteStatuses[i] = 1;
      }

      if(noteStatuses[i+4] == 0 && values[modFinger] == 0 && pitches[i] != 0){
        MIDI.sendNoteOn(pitches[i+4], 127, 1);
        noteStatuses[i+4] = 1;
      }
    }
  }
}
```

```
    }  
  } else{  
    if(noteStatuses[i] == 1){  
      MIDI.sendNoteOff(pitches[i], 0, 1);  
      noteStatuses[i] = 0;  
    }  
  
    if(noteStatuses[i+4] == 1){  
      MIDI.sendNoteOff(pitches[i+4], 0, 1);  
      noteStatuses[i+4] = 0;  
    }  
  }  
}  
  
delay(10);  
}
```