

Sistemas Operacionais Embarcados

Ponto de Controle Final

Controlador MIDI

Igor de Alcantara Rabelo

Matrícula: 14/0143751

Universidade de Brasília

Gama, Brasil

E-mail:rabelo.alcantara.igor@gmail.com

Rodrigo Magalhaes

Matrícula: 14/0031154

Universidade de Brasília

Gama, Brasil

E-mail:rodrigomacr2@gmail.com

I. JUSTIFICATIVAS

Dispositivos MIDI são uma solução de software e hardware para problemas musicais. Sua vantagem se encontra na capacidade de simular diversos tipos de sons e instrumentos com grande facilidade, não sendo necessário possuir ou saber tocar vários tipos de instrumentos para produzir uma música rica e complexa. Controladores MIDI ainda podem oferecer funções de mixagem e edição de áudio durante a gravação, oferecendo ao músico a liberdade de experimentar diversas sonoridades e até mesmo de tocar ao vivo.

O controlador MIDI, como representado na figura abaixo, é uma ferramenta musical que permite aos usuários fazerem diversas coisas em um estúdio de música. É de muita importância para quem trabalha em estúdios musicais pois o MIDI dá um grande poder de edição e sincroniza todo o estúdio em uma fábrica de tracks super eficiente, além de outros benefícios [1].



Figura 1. Controlador MIDI.

II. OBJETIVOS

A intenção com este projeto é construir uma plataforma em hardware para produção de músicas on-the-go. A interface deverá possuir botões para representar cada nota da escala musical e para variação dos parâmetros do sinal (frequência, volume) e deverá suportar diversos sintetizadores.

Considerando o alto preço de Pads MIDI no mercado, toma-se como um dos objetivos deste projeto a tentativa de redução do custo da plataforma enquanto mantendo uma boa qualidade.

III. BILL OF MATERIALS

1. Raspberry Pi 3B+;
2. Cartão de memória;
3. Teclado de computador USB;
4. Display LCD;
5. Saídas para speaker (P2) e computador (USB);
6. Caixa de som Mini Speaker com entrada/saída p2;
7. Softwares utilizados: Raspbian, TiMidity++ e ALSA(Advanced Linux Sound Architecture);

IV. BENEFÍCIOS

Um dos grandes benefícios ao se usar um controlador MIDI é o de acionar sons musicais e tocar instrumentos musicais e também estes controladores podem ser usados para controlar outros dispositivos compatíveis com o MIDI, como luzes de palco, mixers de áudio digital e unidades complexas de efeitos de guitarra [3].

V. DESENVOLVIMENTO

Foram feitas algumas pesquisas para fazer a compra de um teclado MIDI que se encaixasse em um modelo mais próximo do profissional, porém, os custos seriam mais altos e também iria demorar mais para que o teclado chegasse porque teria que fazer o pedido em sites de outro país. Foi utilizado um teclado de computador de modelo ABNT2 mostrado na figura 2, para funcionar de modo que substituisse o teclado MIDI.

Para o desenvolvimento do protótipo inicial, foi utilizado um teclado de computador para testar os botões como se fosse um teclado MIDI. O código desenvolvido foi feito utilizando a linguagem C++ onde foi possível utilizar a biblioteca SDL que fornece uma plataforma simples para gráficos, som, e dispositivos de entrada.

A partir do SDL, pode-se interceptar os eventos do teclado (pressionamento e soltura de botões). Separou-se então quatro linhas do teclado com 12 teclas, suportando um alcance de 4 oitavas incluindo os semitons. Durante o processo de mapeamento entre os botões e as notas, percebeu-se que o layout do teclado pode afetar diretamente o reconhecimento de teclas. O SDL utiliza o layout QWERTY americano como base, que não possui referências para botões de acentuação muito comuns em teclados brasileiros. Felizmente, a biblioteca oferece identificações tanto por Keycode, dada pelo valor do símbolo em UTF-8, quanto por Scancode, dada pela posição física no teclado, sendo as duas utilizadas quando conveniente.

Após resolvido o problema de mapeamento, notou-se que o reconhecimento do clique de múltiplas teclas em simultâneo não era realizado nativamente pela biblioteca. Criou-se então um vetor de armazenamento de estados para as 48 teclas e uma função condicional, semelhante aos eventos de borda de subida e de descida, que as impedia de executar as mesmas linhas de código antes que houvesse uma mudança de estados. Assim que o botão é pressionado, a posição do vetor referente à tecla recebe valor 1, e quando solto o valor volta a 0. Por organização, transferiu-se a função criada para um header SimpleEvent.h, reduzindo aproximadamente 600 linhas do código principal. Ambos os códigos podem ser vistos na referência 1.

Para a segunda etapa de desenvolvimento do projeto, implementou-se o envio de mensagens MIDI, adicionando funcionalidades básicas ao projeto. Utilizando-se a biblioteca RtMidi.h, criou-se um canal de comunicação entre o driver de áudio do Raspbian (ALSA) e um sintetizador open source third party (TiMidity++), possibilitando a reprodução de áudio.

Seguindo o protocolo de mensagens MIDI [6], definiu-se as funções responsáveis por ligar e desligar as notas musicais, aumentar e diminuir o volume, variar a frequência das teclas e alterar o instrumento sintetizado. As mensagens são basicamente compostas por entre 2 e 3 bytes, sendo o primeiro

byte de status (informa a operação a ser realizada) enquanto os remanescentes são parâmetros dessa instrução.

O layout do teclado em relação às funções pode ser visto a seguir:



Figura 2. Layout para teclado do tipo ABNT2.

VI. RESULTADOS

Para a fase inicial do projeto, utilizou-se a impressão do nome das notas e seu estado de ativação (pressionado ou solto) para verificar o funcionamento do código (Figura 3). Verificou-se que o reconhecimento de teclas simultâneas funcionou corretamente e o vetor de armazenamento de estados foi utilizado para a reprodução das formas de onda. Com a função KeyboardWatchdog, da biblioteca SimpleEvent.h, adaptada para receber apenas uma struct como argumento será possível criar uma thread para rodar a interceptação de teclas paralelamente a função que reproduz áudio, esperando-se assim a redução do tempo de resposta do programa.

```
rodrigo@rodrigo-270E5J-2570EJ: ~/Área de Trabalho/RasPiano
Arquivo Editar Ver Pesquisar Terminal Ajuda
rodrigo@rodrigo-270E5J-2570EJ:~/Área de Trabalho/RasPiano$ g++ main.cpp `pkg-con
fig --cflags --libs sdl2`
rodrigo@rodrigo-270E5J-2570EJ:~/Área de Trabalho/RasPiano$ ./a.out
C0 pressed
C#0 pressed
D0 pressed
D#0 pressed
D0 released
C#0 released
D#0 released
C0 released
E0 pressed
F#2 pressed
G#3 pressed
G#3 released
F#2 released
E0 released
D#2 pressed
E2 pressed
E2 released
F2 pressed
F2 released
F#2 pressed
F#2 released
D#2 released
```

Figura 3. Reconhecimento de teclas.

Para a segunda fase do projeto, conclui-se a interface de áudio. Agora, ao invés de imprimir as notas no terminal, o programa envia mensagens MIDI ao sintetizador que agora reproduz o som em tempo real. Ao ligar e desligar as notas, passa-se um valor específico para a frequência e outro para o volume. Neste caso, nenhum parâmetro desta mensagem pode ser alterado retroativamente, o que significa que para mudar

de volume ou a frequência base da nota, deve desligá-la e ligá-la novamente.

A função de bend de frequência, por ser uma mensagem separada da instrução de ativação de notas, é capaz de distorcer a frequência da nota em tempo real (retroativamente). Adicionou-se também a opção de troca de instrumentos. Utilizando-se das combinações numéricas de 000 a 127 do Keypad, tornou-se possível selecionar entre todos os instrumentos disponíveis na SoundFont para teste e para selecionar todos os 128 timbres possíveis. Para selecionar o timbre desejado, foi utilizado uma espécie de menu para a troca usando as teclas do próprio teclado MIDI.

Por fim foi adicionado uma função de looping. Ao ser reproduzido um conjunto de notas usando o looping é possível fazer com que as notas sejam gravadas e reproduzidas novamente da mesma forma que foi tocada antes.

Na figura 4 está sendo apresentado um fluxograma de cada função do teclado MIDI. Nesse fluxograma estão contidas as funções de instrumentos, volume, notas, frequência e looping. Nele também está explicando o algoritmo que foi usado para o funcionamento do teclado MIDI usando a RaspBerry Pi.

DISCUSSÕES E CONCLUSÃO

Em conclusão, foram efetivamente implementadas as funções de toque polifônico, variação de volume, distorção de frequência, troca de instrumentos e loop. Em relação aos gastos, um protótipo com visual mais profissional custaria por volta do dobro do preço do projeto em seu estado atual (teclado comum + RaspBerry Pi 3B+). Ao início do projeto, esperava-se problemas de tempo de resposta perceptível entre o pressionar das teclas e a saída do áudio. Felizmente, estes problemas não foram observados.

Como propostas de melhorias, a inclusão de uma interface de display e a solução de alguns problemas relacionados ao loop (habilidades de manter o instrumento utilizado durante a gravação e de limpar o loop armazenado em memória a fim de se iniciar outro).

No geral, o projeto excedeu as perspectivas, sendo adicionadas funções muito mais complexas que as planejadas inicialmente.

Por fim, a lista de instrumentos utilizados e seu código de ativação são apresentados em anexo na figura 5.

VII. REFERÊNCIAS

- [1] RasPiano:
<https://github.com/RodrigoMac/Sistemas-Embarcados/tree/master/RasPiano>
- [2] SDLwiki:
<https://wiki.libsdl.org/FrontPage>
- [3]Raspberry Pi Midi-Keyboard:
<https://www.youtube.com/watch?v=mpeTHLr1vIc>
- [4]ALSA(Advanced Linux Sound Architecture):
https://www.alsa-project.org/wiki/Main_Page
- [5]Timidity:
<https://wiki.archlinux.org/index.php/Timidity>
- [6]Summary of MIDI Messages:
<https://www.midi.org/midi/specifications/item/table-1-summary-of-midi-message>

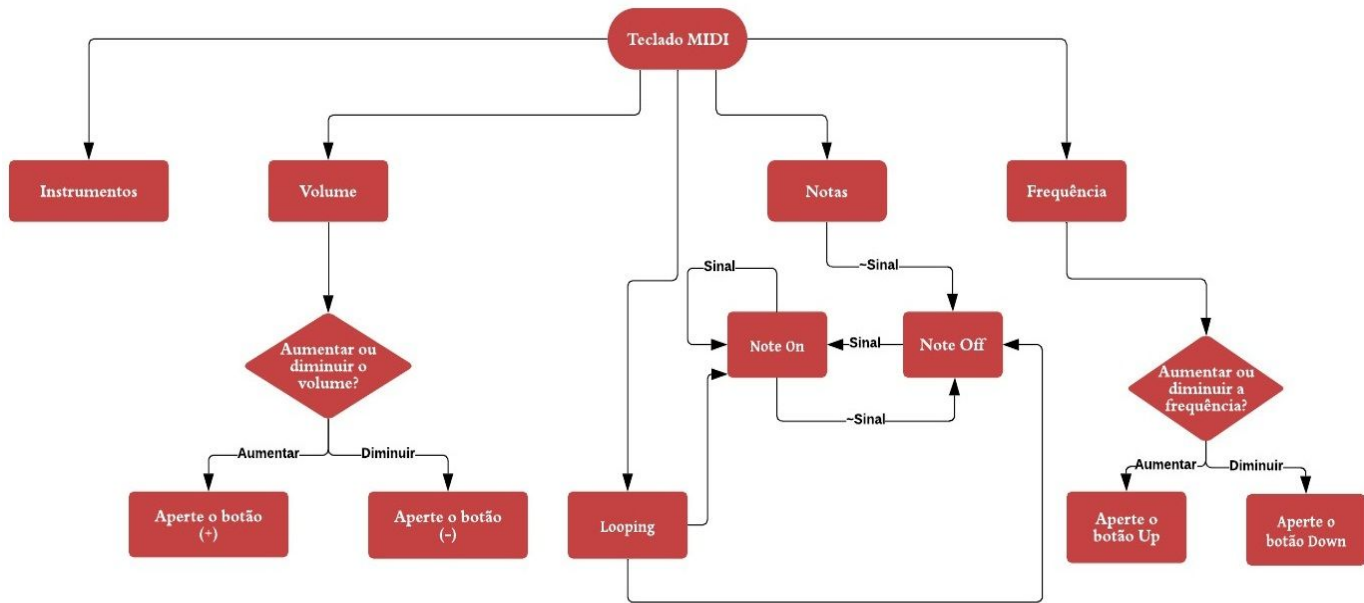


Figura 4. Fluxograma do teclado MIDI.

Piano Family 1 Acoustic Piano 2 Bright Piano 3 ElecGrand 4 Honky Tonk 5 Elec Piano 1 6 Elec Piano 2 7 Harsichord 8 Clavichord	Bass Instruments 33 Acoustic Bass 34 Finger Bass 35 Pick Bass 36 Fretless Bass 37 Slap Bass 1 38 Slap Bass 2 39 Synth Bass 1 40 Synth Bass 2	Reed Instruments 65 Soprano Sax 66 Alto Sax 67 Tenor Sax 68 Baritone Sax 69 Oboe 70 English Horn 71 Bassoon 72 Clarinet	Synthesized Effects 97 Rain 98 Soundtrack 99 Crystal 100 Atmosphere 101 Brightness 102 Goblins 103 Echoes 104 Sci-Fi
Chromatic Percussion 9 Celesta 10 Glockenspiel 11 Music Box 12 Vibraphone 13 Marimba 14 Xylophone 15 Tubular Bells 16 Dulcimer	Strings, Harp and Timpani 41 Violin 42 Viola 43 Cello 44 Contra Bass 45 Tremelo Strings 46 Pizzicato 47 Harp 48 Timpani	Other Woodwind Instruments 73 Piccolo 74 Flute 75 Recorder 76 Pan Flute 77 Blown Bottle 78 Shakuhachi 79 Whistle 80 Ocarina	Ethnic Instruments 105 Sitar 106 Banjo 107 Shamisen 108 Koto 109 Kalimba 110 Bagpipe 111 Fiddle 112 Shanai
Organ Family and Accordians 17 Drawbar Org 18 Percussive 19 Rock Organ 20 Church Organ 21 Reed Organ 22 Accordion 23 Harmonica 24 Tango Accdn	Ensemble Voices 49 Strings 1 50 Strings 2 51 Synth Strings 1 52 Synth Strings 2 53 Choir Aahs 54 Voice Oohs 55 Synth Voice 56 Orchestra Hit	Synthesized Lead Sound 81 Square Wave 82 Sawtooth 83 Calliope 84 Chiffer 85 Charang 86 Voice Lead 87 Root+Fifth 88 Bass Lead	Percussive Instruments 113 Tinkle Bell 114 Agogo 115 Steel Drums 116 Woodblock 117 Taiko Drum 118 Melodic Tom 119 Synth Drum 120 Reverse Cymbal
Guitars 25 Acoustic Gtr 26 Steel Guitar 27 Jazz Guitar 28 Clean Elect Gtr 29 Muted Guitar 30 Overdriven Gtr 31 Distorted Gtr 32 Harmonics	Brass Instruments 57 Trumpet 58 Trombone 59 Tuba 60 Muted Trumpet 61 French Horn 62 Brass Section 63 Synth Brass 1 64 Synth Brass 2	Synthesized Pads 89 New Age 90 Warm Pad 91 Polysynth Pad 92 Choir Pad 93 Bowed Pad 94 Metallic Pad 95 Halo Pad 96 Sweep Pad	Sound Effects 121 Fret Noise 122 Breath Noise 123 Sea Shore 124 Bird Tweet 125 Telephone 126 Helicopter 127 Applause 128 Gun Shot

Figura 5. Lista de instrumentos implementados no teclado MIDI.