**Alunos:**

Danilo Matos Valente 2017002490

Gabriel Portela Saliba 2017002466

Joaquim Barreto 2017430565

**Turma:** T??? **Disciplina:** Física Experimental EO

**Projeto: Controlador Musical Binário**

**Introdução** (Motivação, Objetivos (De projeto final e de protótipo))

A presença de controladores no repertório de um produtor musical, de um hobbista ou mesmo de um estudante está aumentando, com instrumentos que vão desde teclados a flautas e guitarras eletrônicas. A ausência de instrumentos desse tipo que ser operados com uma mão apenas e sem a presença de uma superfície de apoio nos levou a ter a ideia proposta.

Como objetivo, desejamos criar um instrumento musical eletrônico de fácil uso e configuração, que possa ser utilizado com uma mão e que enviará comandos MIDI de modo que o receptor sintetize o som desejado. Ainda que a configuração binária (cada botão equivale a uma casa binária), isto poderá ser alterado depois.

Nosso protótipo consiste em um controle *Wiichuck* (com 2 botões, um joystick e um acelerômetro X-Y) modificado de modo a receber 3 botões extras, estando todos ligados a um microcontrolador Arduino, carregado via USB. A combinação dos 5 botões e o manuseio do joystick no eixo X (no protótipo atual, as demais entradas não foram utilizadas) realizam a emissão de frequências por um buzzer.

**Desenvolvimento**

Material Utilizado

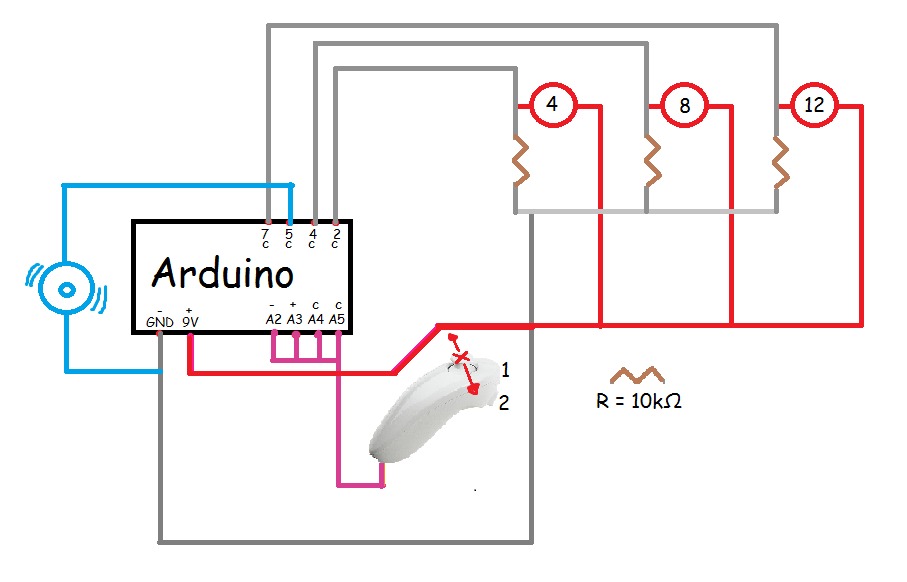
* Arduino UNO, com protoboard, cabos *jumper* e cabo USB
* Controle *nunchuck* (Controle secundário, utilizado no console de jogos *Wii*)
* 3 Botões tipo *pushbutton*
* Buzzer *piezo*
* Cabos, fios e ferramentas de solda

Funcionamento

Após o Arduino ser conectado a uma fonte, será possível verificar que os botões emitem sons no Buzzer. A configuração é dada por:

* Cada botão no corpo externo do aparelho emite uma quantidade de semitons. De cima para baixo: 1, 2, 4 e 8 semitons. A combinação dos botões resulta em uma soma diferente. Selecionando, de cima para baixo, por exemplo, o primeiro e o terceiro botões, haverá uma soma igual a 5.
* O botão acima do Joystick emite 12 semitons, equivalente a uma oitava. Ou seja, segurá-lo enquanto outros botões são selecionados irá aumentar uma oitava da nota a ser tocada. Selecionando o primeiro, o terceiro e este botão (quinto), por exemplo, haverá uma soma igual a 17.
* O joystick em seu eixo X, ao ser deslocado para a esquerda ou para a direita, resulta, respectivamente, em uma subtração ou soma de, no máximo, 2 semitons. Útil para alternar entre notas próximas ou realizar *legato* (uma nota “grudada” à próxima).
* Por fim, é configurado no código uma nota inicial. No protótipo, isso foi feito tal que o primeiro semitom gerado equivale ao B4 (Si na quarta oitava). Assim, selecionando, por exemplo, o primeiro, o terceiro e o quinto botões, será gerada a nota F#5 (Fá na quinta oitava).

Circuito



O circuito é bem simples. O Arduino, ligado a uma fonte (Bateria 9V ou via USB) recebe os comandos dos botões e do *nunchuck*, processa-os e envia uma resposta ao buzzer. É importante ressaltar o uso de um adaptador que liga a saída do *nunchuck* ao arduino, envialndo comandos que são (botões, eixo x do joystick) ou não (aceletrômetro, eixo y do joystick) utilizados.

Software

Resumidamente, o código, escrito em C++ e com a biblioteca *Arduino.h*, realiza uma leitura digital constante de cada um dos botões e do joystick. Existe uma estrutura vetor que armazena, sequencialmente, 89 frequências possíveis para o *buzzer* executar.

|  |
| --- |
| *notas() {*  *nota[0] = 31;*  *…*  *nota[46] = 440; (Este é o Lá da quarta oitava)*  *…*  *nota[88] = 4978*  *}* |

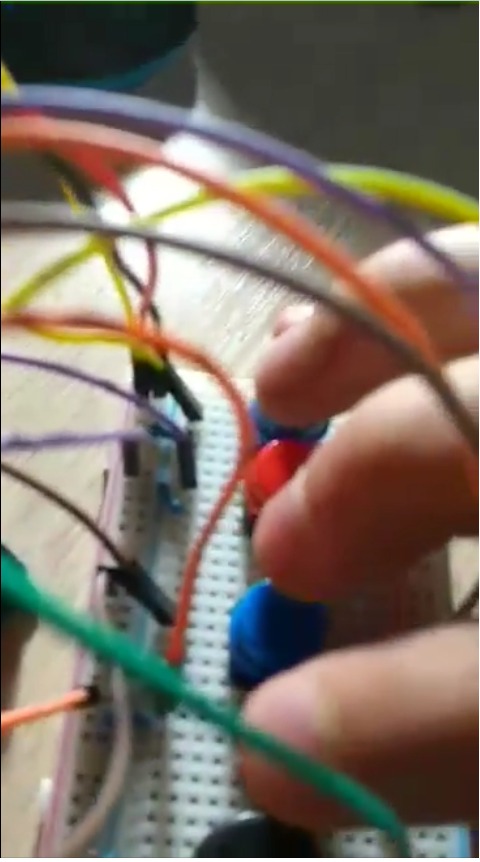
Conforme explicado acima, cada botão possui um peso. Uma variável *soma* armazena a soma de cada botão multiplicado pelo seu peso, uma variável *bend* armazena a modificação da nota provocada pelo *joystick* e uma variável *inicio* possui a nota anterior ao primeiro semitom executável. Assim, temos, como pseudocódigo:

|  |
| --- |
| *soma = botao1 \* 1 + botao2 \* 2 + botao4 \* 4 + botao8 \* 8 + botao12 \* 2;*  *bend = gerarVariacao(comando\_joystick\_x); // Gera inteiro entre [-2, 2]*  *inicio = 47; // Nota Bb4, anterior à nota B4*  *tocarNota(nota[ soma + bend + inicio ]); // Toca a nota requisitada* |

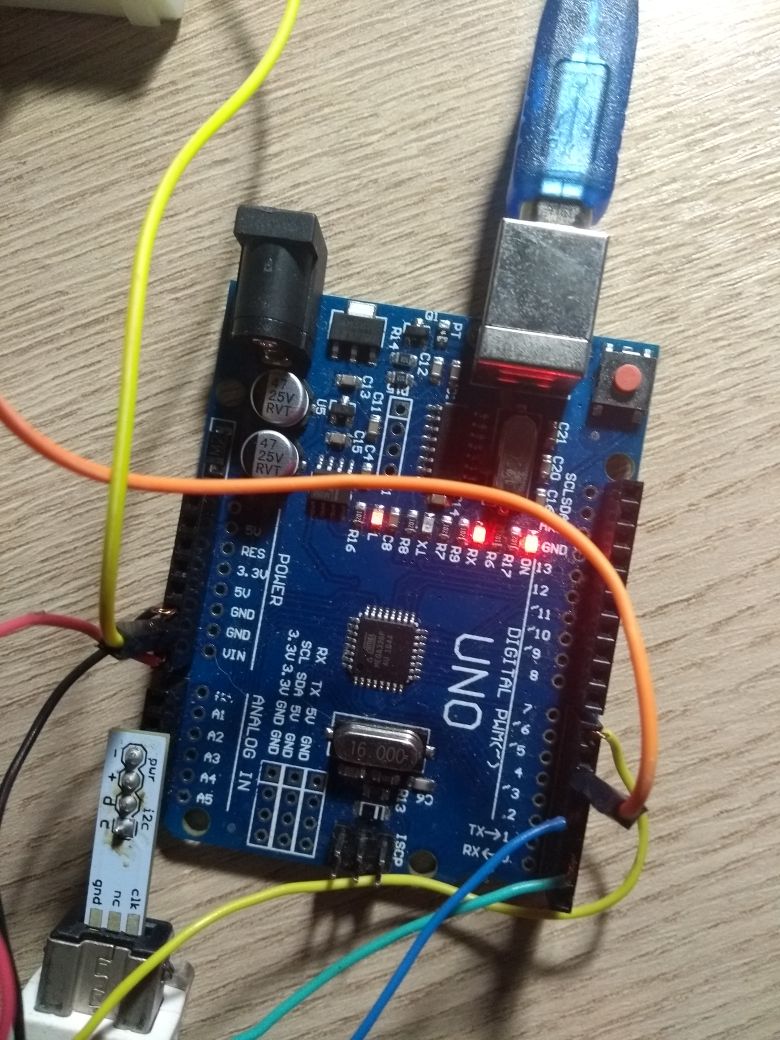
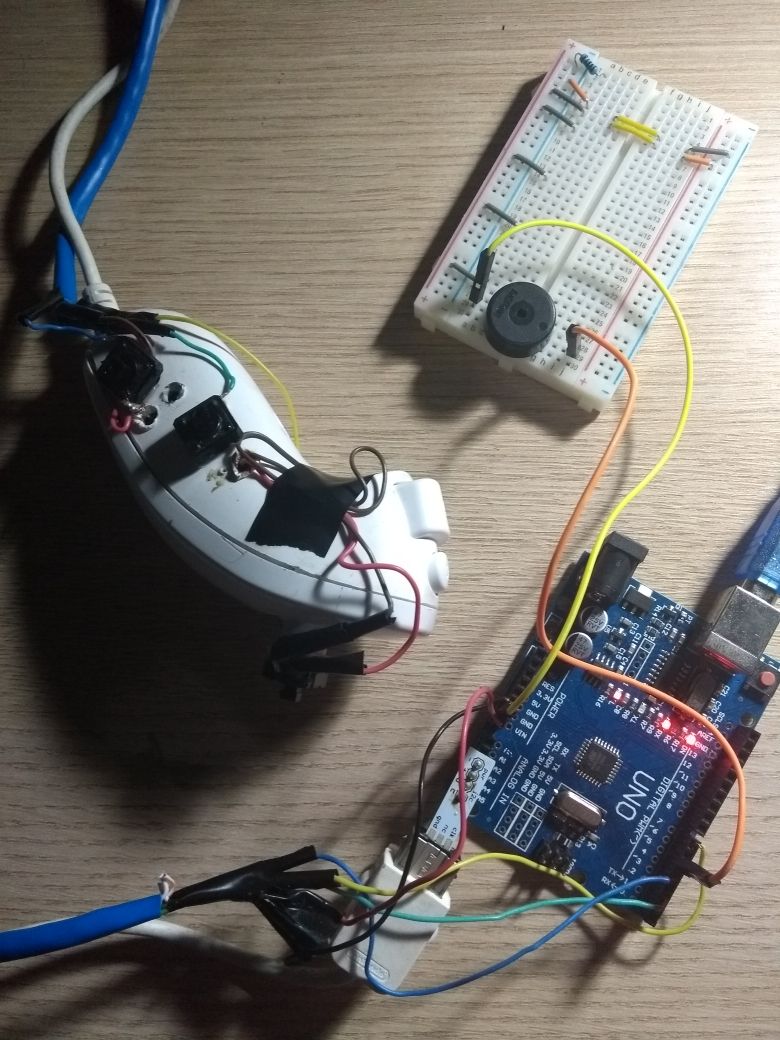
Além destes procedimentos, diversos outros foram utilizados para melhorar a estabilidade e o funcionamento do controle. Dentre eles, uma função que detecta transição entre notas, a fim de suavizá-las, e uma função que busca normalizar os *inputs* analógicos do joystick, a fim de tornar a mudança do tom mais suave.

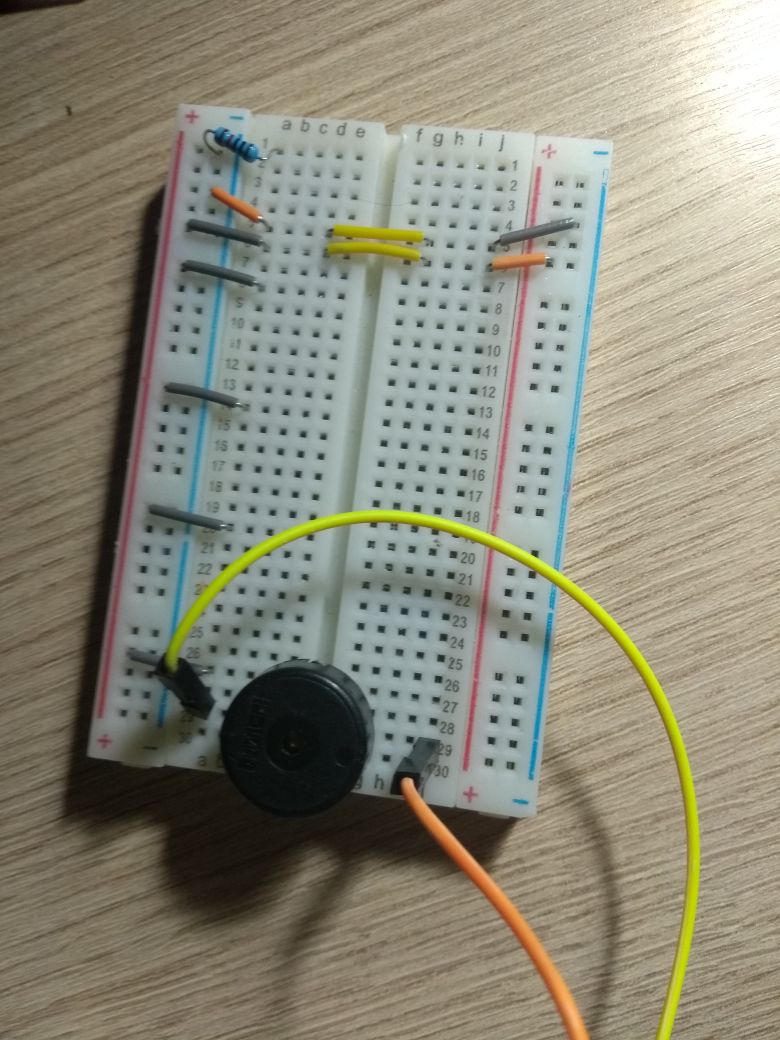
Mídia e Exemplos

De início, nosso projeto consistia em quatro botões (sem o botão que adiciona uma oitava) e o buzzer, dispostos em uma protoboard.



O protótipo final consiste na adição de botões ao corpo do *nunchuck*, utilizando seu joystick e seus botões. Aqui temos o sistema completo, e um *zoom* na configuração do Arduino.



As imagens a seguir mostram o buzzer em uma protoboard e posições diferentes do controle, assim como o seu interior, a fim de ver as conexões internas e externas presentes.

**Conclusões** (Resultado final, dificuldades e aprendizados, possíveis melhorias)

Ao final, o controle mostrou-se funcional, ainda que não muito confortável de usar. Conseguimos rapidamente reproduzir algumas melodias, como a canção folclórica *greensleeves*, o trecho principal de *Ode to Joy* (*Ludwig van Beethoven*) e as escalas maiores e menores a partir da primeira nota, B4.

O projeto nos forneceu amplo conhecimento de eletrônica básica geral, tanto teórica como prática, além de programação em Arduino. As maiores dificuldades enfrentadas compõem a elaboração de funções, citadas acima, com o intuito de tornar mais agradável o uso das notas, assim como os procedimentos práticos de eletrônica (soldagem e manuseio de fios, botões e do próprio controle). Ademais, por se tratar de um protótipo inicial, não tomamos tanto cuidado no quão ergonômico tornou-se o controle.

Futuramente, muitos procedimentos podem ser realizados para tornar o controle mais confortável, customizável e fácil de usar. Dentre eles, a produção de um corpo e de botões apropriados (utilizando, por exemplo, uma impressora 3D), usar conexão bluetooth, adicionar *velocity* às notas (a força aplicada ao botão determina a intensidade da nota), elaborar alguma forma de prender o controle à mão sem ter de pressioná-lo (utilizando, por exemplo, um velcro) e, principalmente, fazê-lo transmitir comandos MIDI em vez de acionar um *buzzer*.