

Projeto 07

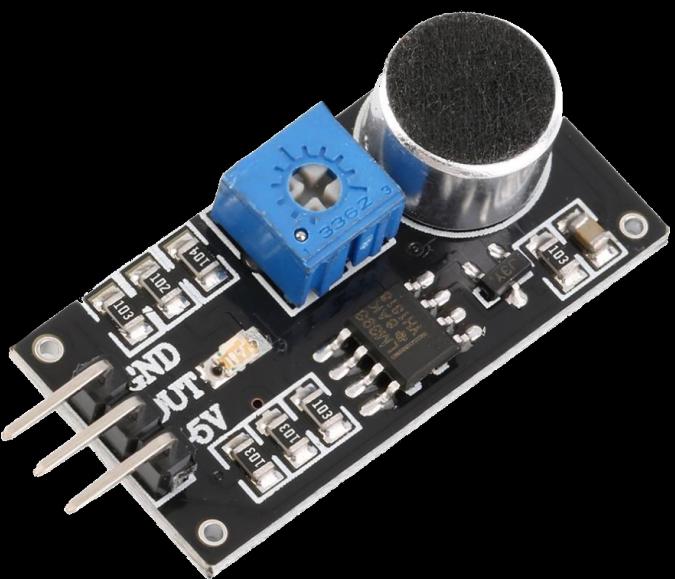
Controle Sonoro – Prática

Jan K. S. – janks@puc-rio.br

ENG1419 – Programação de Microcontroladores

Testes Iniciais

19



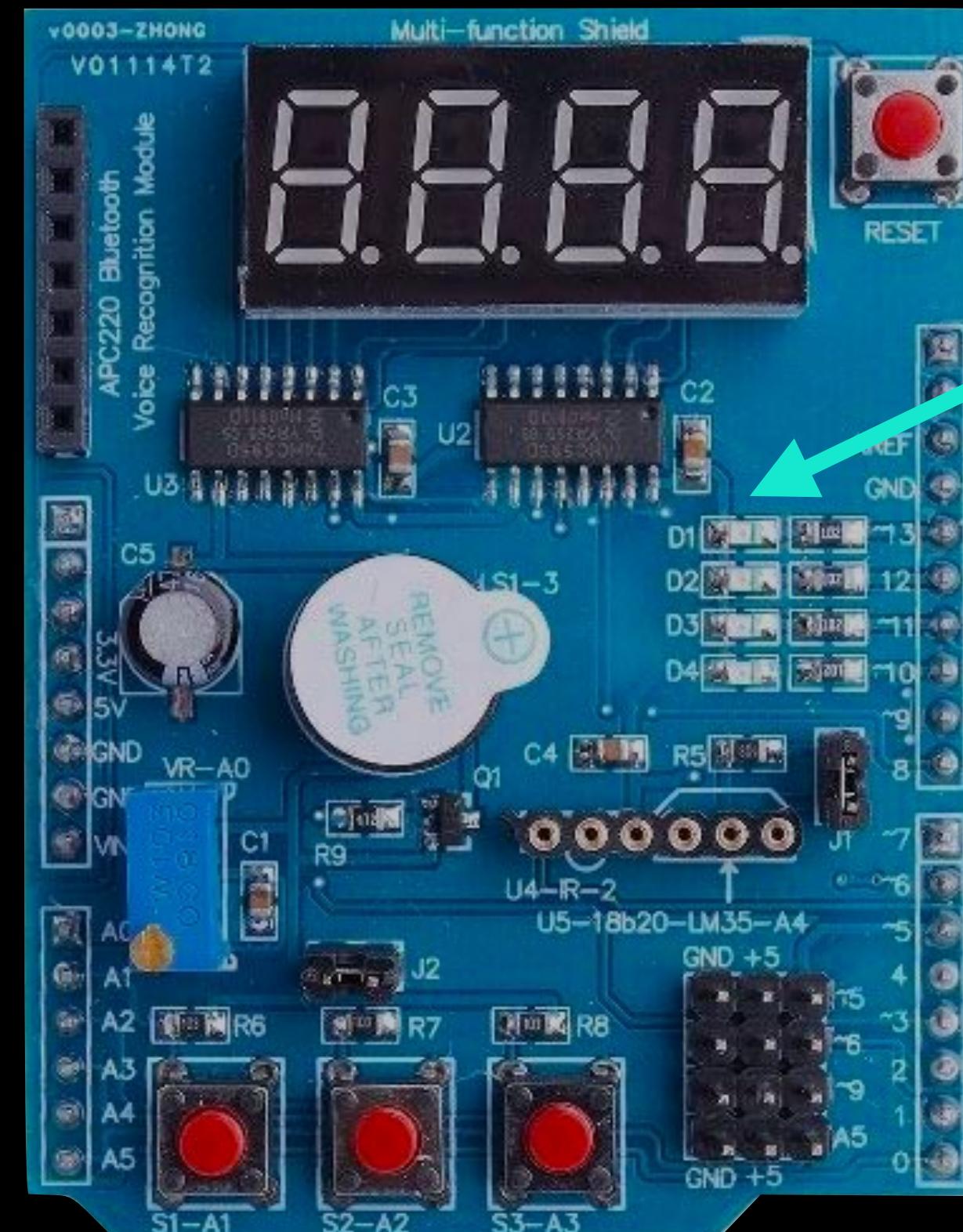
20,21



A5 5



pinos 4,7 e 8



pinos 13,12,11 e 10

pinos A1, A2 e A3

Pinos Usados pelos Componentes



Testes Iniciais

Ao apertar o Botão 1, **toque a campainha 1 vez**, com uma frequência 440Hz durante 500ms.

↪ DICA: use a GFButton.

Ao apertar o Botão 2, **toque a campainha com uma frequência 220Hz continuamente**. Ao soltar, **pare de tocar a campainha**.

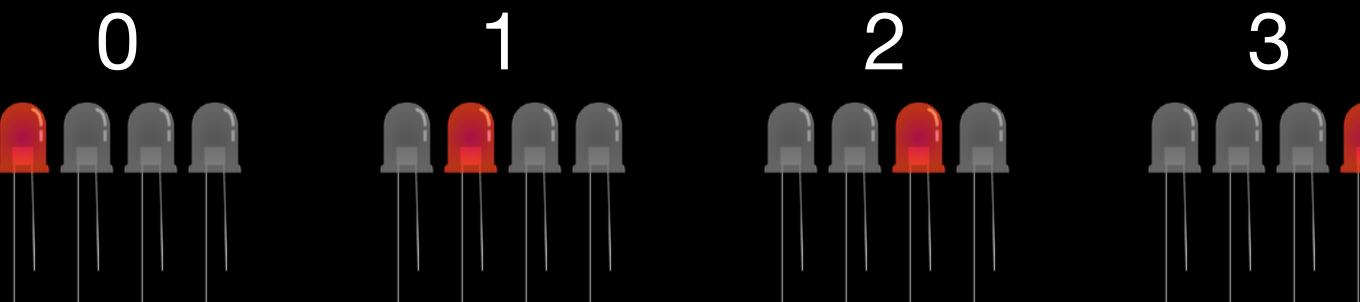
↪ DICA: use a GFButton e a função noTone.

Conte quantas vezes houve um estalar de dedos, e **exiba essa contagem no display de 7 segmentos**.

↪ DICA: use as funções attachInterrupt e millis.

Acenda os LEDs de acordo com o valor do **módulo do resto da divisão por 4 da posição do encoder**.

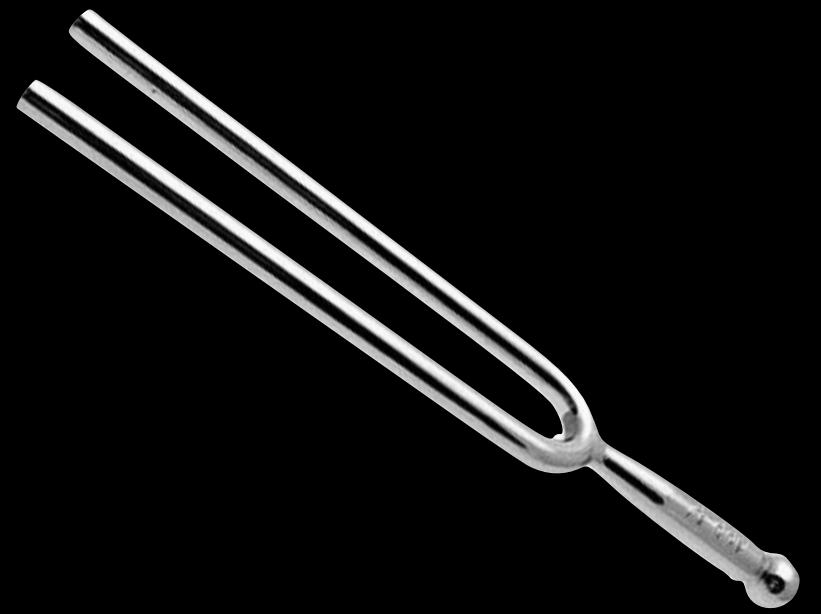
↪ DICA: use a getPosition e um for no loop. Pesquise como fazer o módulo (valor absoluto) em Arduino.



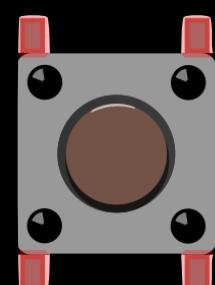
Implementação



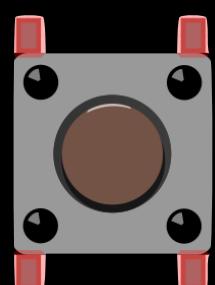
Controle Sonoro Musical



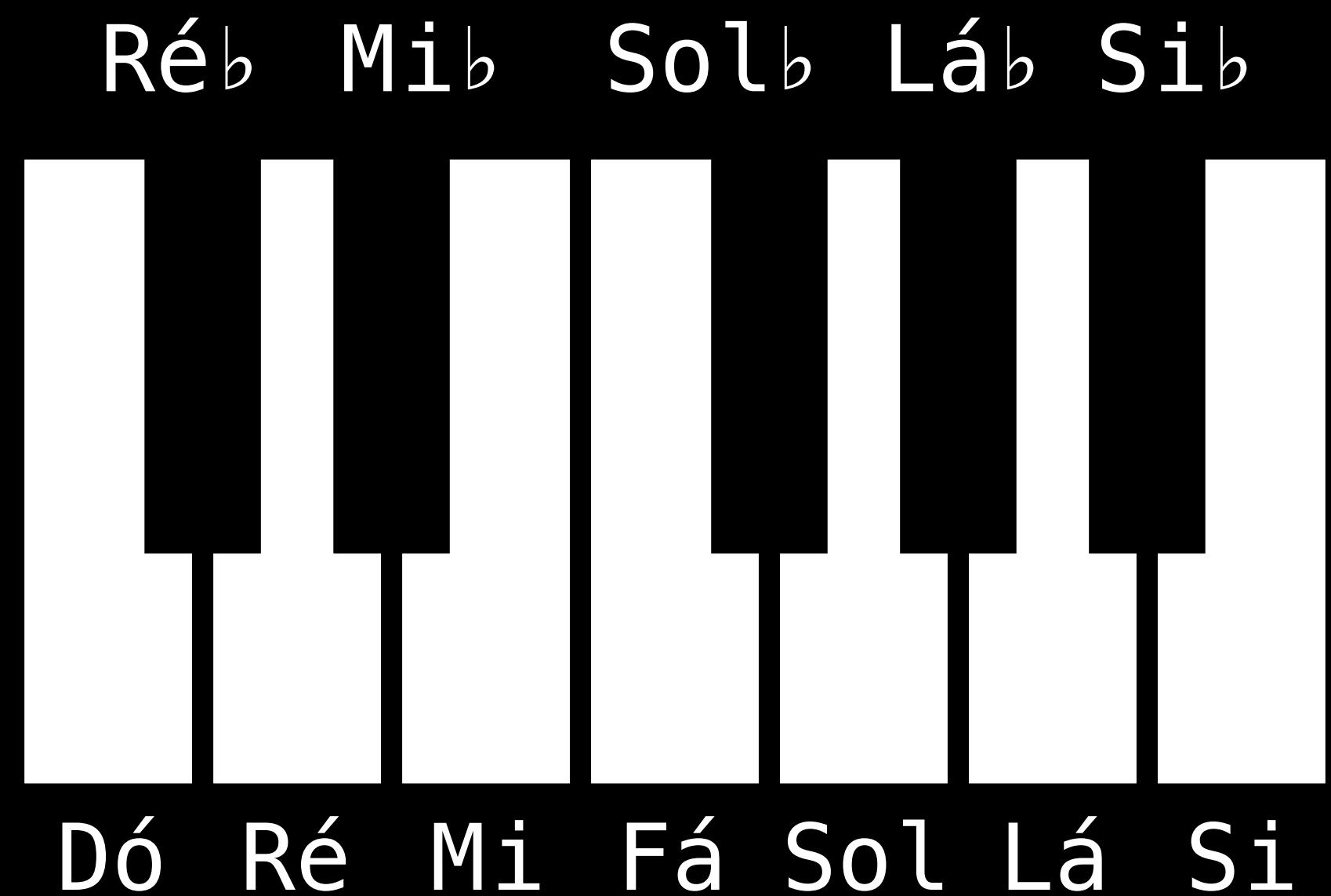
Botão 1
modo Afinador



Botão 2
modo Metrônomo

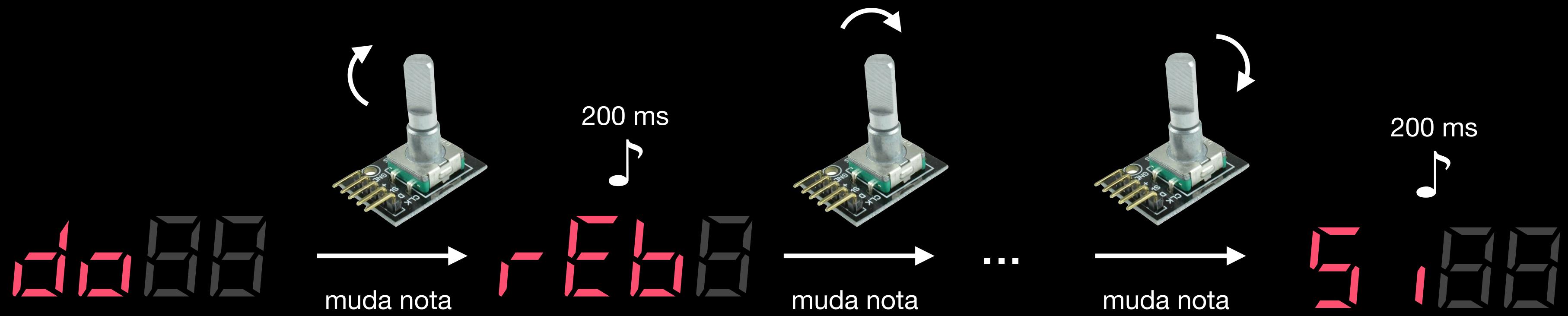


Afinador e Metrônomo



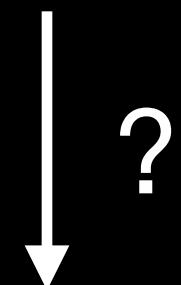
```
char* nomeDasNotas[] = {"D0 ", "REb", "RE ", ...};  
int frequencias[] = {131, 139, 147, ...};
```

Botão 1



Controle do Afinador via Botão 1 e Potenciômetro

```
int posicao = encoder.getPosition()  
... → -4 → -3 → -2 → -1 → 0 → 1 → 2 → 3 → 4 → ...
```



```
int notaAtual  
0 → 1 → 2 → 3 → ... → 10 → 11
```

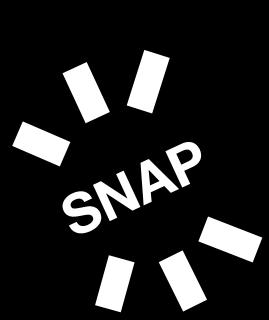
se posição > posição anterior
aumenta nota atual (parando em 11)
se posição < posição anterior
diminui nota atual (parando em 0)



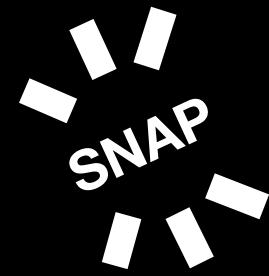
Metrônomo para Marcação de Ritmo



Botão 2



instante1



instante2



```
unsigned long periodoEmMs = instante2 - instante1;
```

```
int batidasPorMinuto = 60000 / periodoEmMs;
```



BB75 BIP BIP

Botão 2: Inicia Metrônomo Digital com Base nos Estalos



Implementação

Ao apertar o Botão 1, entre no modo "Afinador", tocando a nota atual. Ao soltar, pare de tocar a nota.

↪ DICA: use uma variável global para indicar o índice da nota atual nos vetores.

Ao mexer no encoder, mude a nota atual, toque-a por 200 ms e exiba seu nome no display. Não ultrapasse a nota mais baixa ou a mais alta.

↪ DICA: varie o índice da nota atual para cima ou para baixo de acordo a mudança de posição do encoder.

Ao apertar o Botão 2, entre no modo metrônomo: detecte dois estalos consecutivos e exiba o período entre eles (em batidas por minutos) no display.

↪ DICA: use uma segunda variável global para salvar o instante do primeiro estalo.

Usando o intervalo entre os dois estalos, toque um bip periódico. Pare de tocá-lo ao apertar o Botão 1 ou o 2.

↪ DICA: use a millis dentro do loop.

Aperfeiçoamento



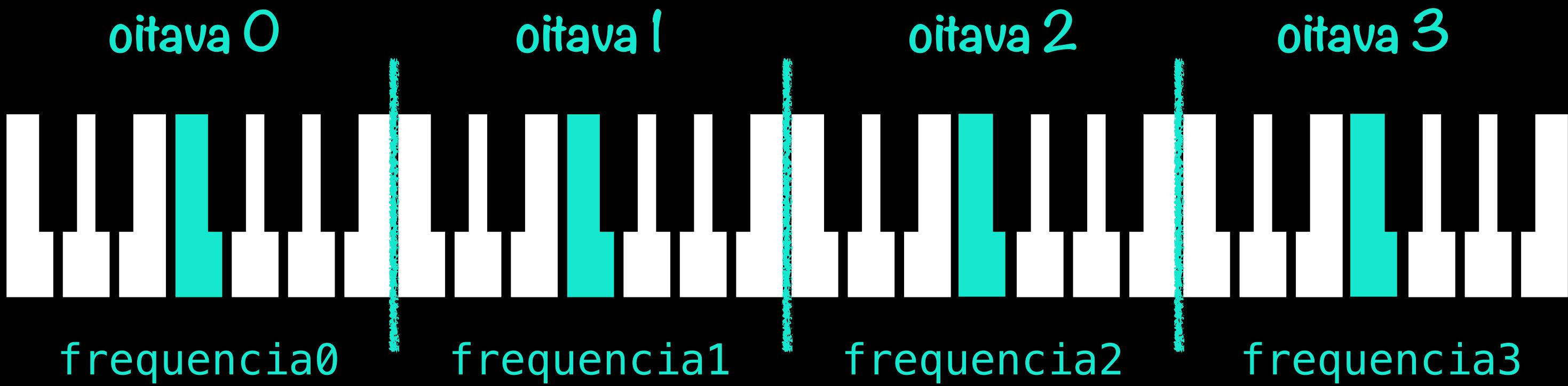
p07b_implementacao.ino

cópia
----->



p07c_aperfeicoamento.ino

Cópia do Código da Implementação para o Aperfeiçoamento

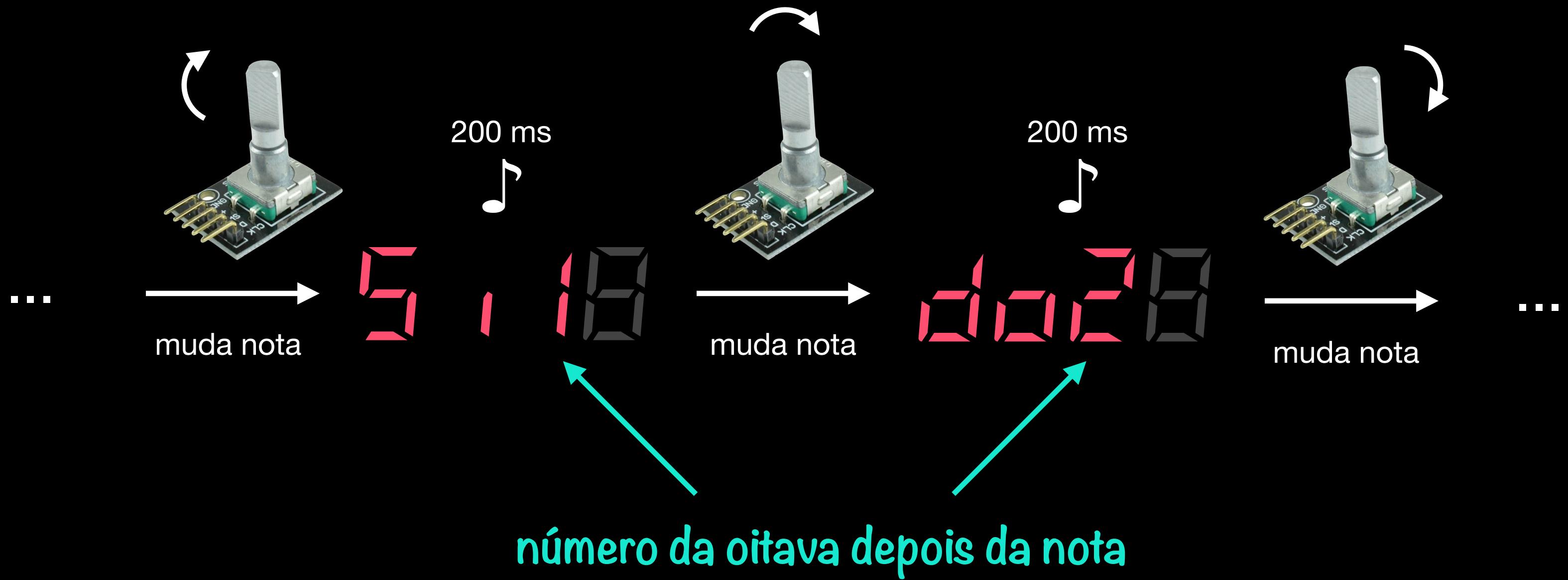


$$\text{frequencia}_N = \text{frequencia}_0 \times 2^N$$

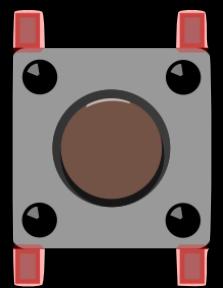


`frequenciaN = frequencia0 * pow(2, N);`

Múltiplas Oitavas



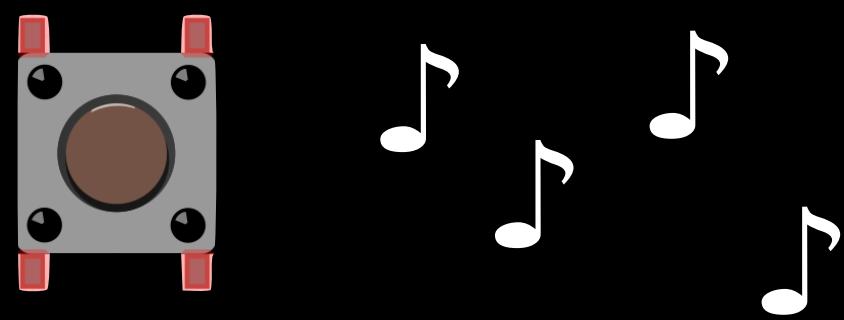
Exibição da Oitava no Display



Botão3



Caixinha de Música no Botão 3



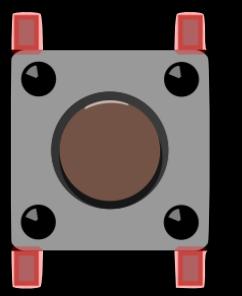
Botão3

```
char* nomeDasNotas[] = {"D0 ", "REb", "RE ", ...};  
int frequencias[] = {131, 139, 147, ...};
```



```
int indicesDeNotaDaMusica[] = {7, 2, 0, 11...};  
int oitavasDaMusica[] = {0, 1, 1, 0, ...};  
int intervalosEntreNotas[] = {1000, 1000, 167, 167...};
```

Dados da Música



Botão3



toca primeira nota
registra instante

loop do Arduino:

se já passou o tempo esperado desde a nota anterior:

toque a nota atual

exiba a nota atual no display

incremente o índice da nota atual



Ao mexer no encoder no modo Afinador, **caminhe entre as notas de todas as quatro oitavas**, sem ultrapassar a nota mais baixa da primeira oitava ou a mais alta da última.
↪ DICA: use uma outra variável global para guardar o valor da oitava atual.

Mostre o número da oitava ao lado da nota no display.

↪ DICA: pesquise sobre a função `sprintf` para concatenar uma string com um número inteiro.

Ao apertar o Botão 3, entre no modo caixinha musical, **toque a primeira nota** da música (durando 500ms) **e registre o instante** de tempo atual.

Aperfeiçoamento

No loop do **Arduino**, verifique se já se passou o tempo esperado desde a nota anterior, **toque a próxima nota e exiba o nome dela** (com a oitava) no display.

↪ DICA: use a função `millis` dentro do loop. Você também pode criar uma função auxiliar que toque e exiba a nota, em vez de repetir o código dos itens anteriores.

Desafio Extra



p07c_aperfeicoamento.ino

cópia
----->

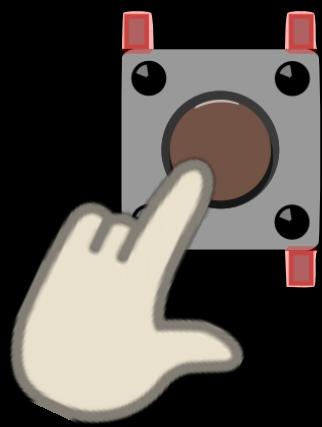


p07d_desafio.ino

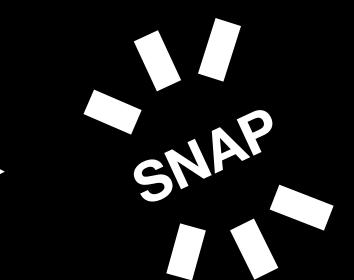
Cópia do Código do Aperfeiçoamento para o Desafio

gravação

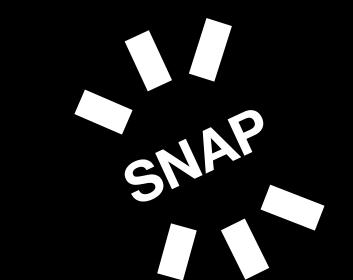
Botão 2



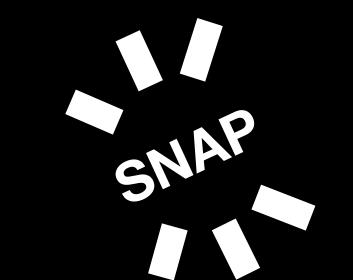
→
0.2s



→
0.7s



→
1.1s

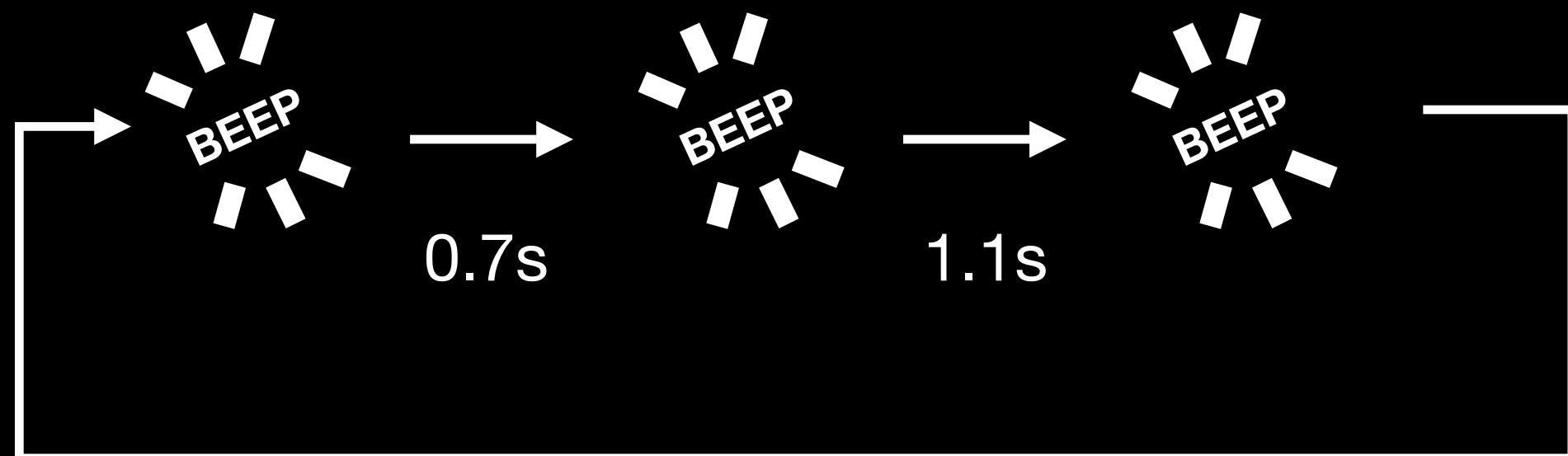


→
0.3s



Botão 2

reprodução



0.2s + 0.3s

```
unsigned long intervalos[100];
```

```
int totalDeEstalos;
```

```
int indiceDoEstaloAtual;
```

Gravação e Reprodução de Padrão de Estalos



Desafio Extra

Ao apertar o Botão 2, **detecte uma sequencia de estalos** e armazene o intervalo de tempo entre cada um. Ao soltar o Botão 2, **imprima esses intervalos**.

↪ DICA: armazene os valores em um vetor bem grande, e conte quantos aconteceram. O primeiro valor deve ser a soma do primeiro intervalo com o último.

Após soltar o Botão 2, **reproduza essa sequencia de estalos com beeps da campainha** (440 Hz durando 50ms) dentro do **loop**.

↪ DICA: crie uma nova verificação de tempo função **millis**.



janks.link/micro/projeto07.zip

Material do Projeto 07