



Projeto 07

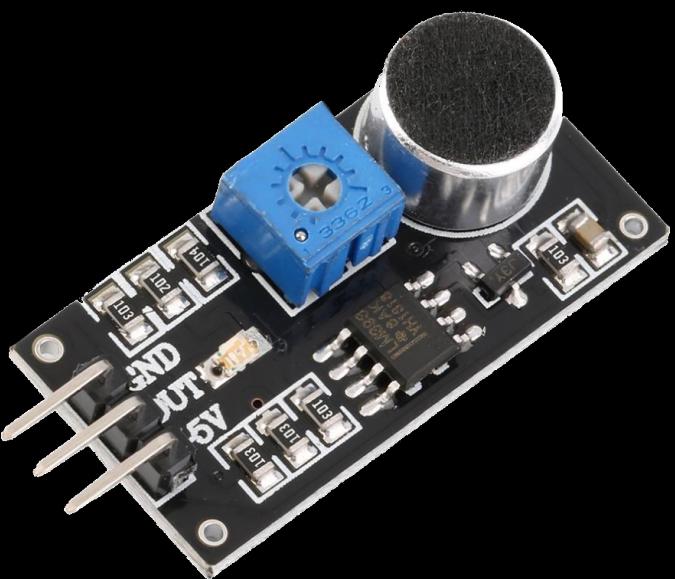
Controle Sonoro – Prática

Jan K. S. – janks@puc-rio.br

ENG1419 – Programação de Microcontroladores

Testes Iniciais

19



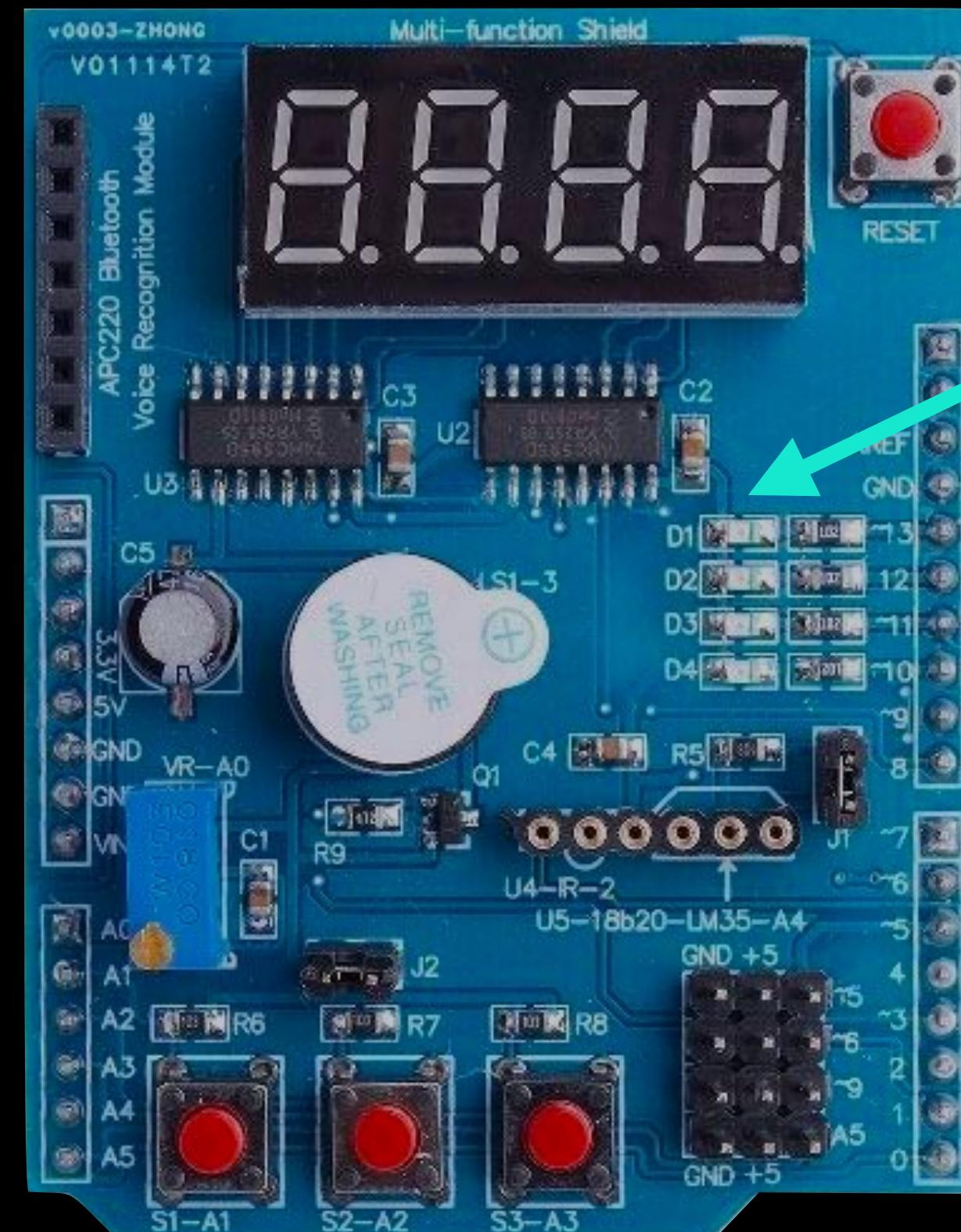
20,21



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**.

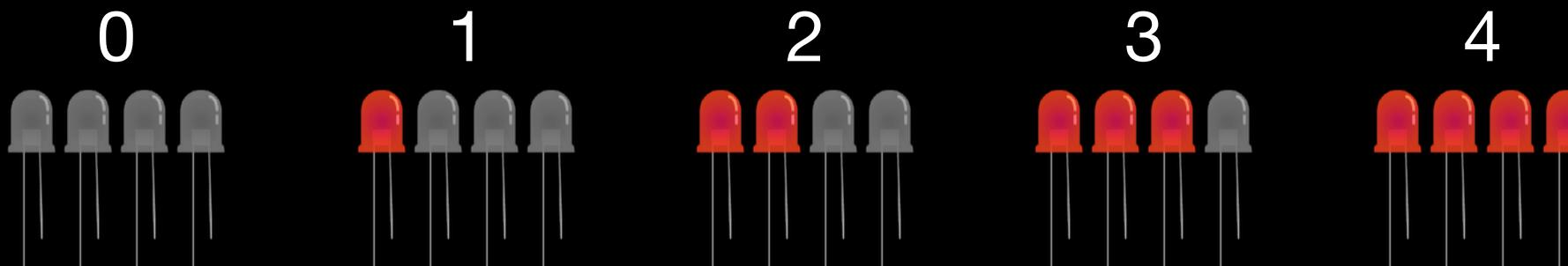
↪ DICA: use a GFButton e a função noTone.

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 5 da posição do encoder**.

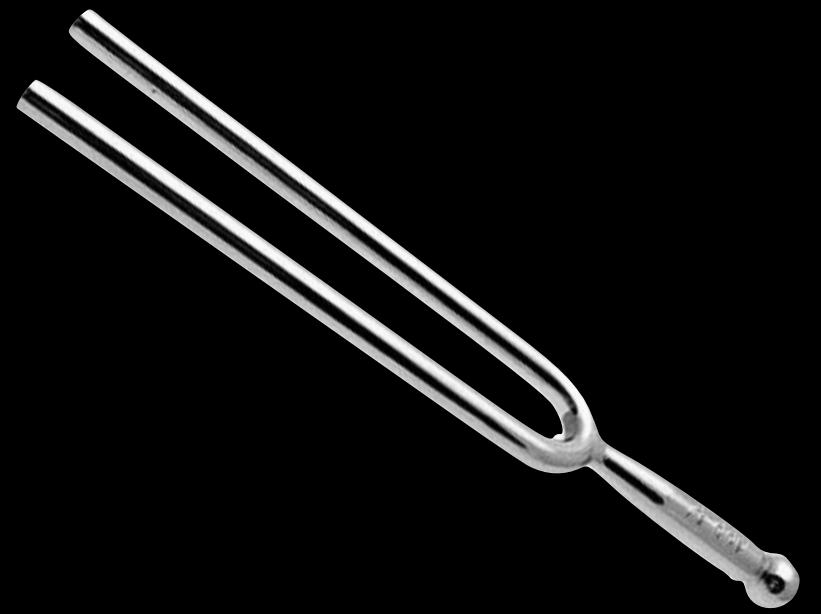
↪ DICA: use a getPosition no loop.



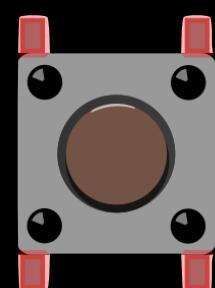
Implementação



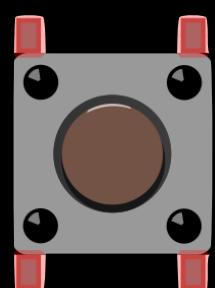
Controle Sonoro Musical



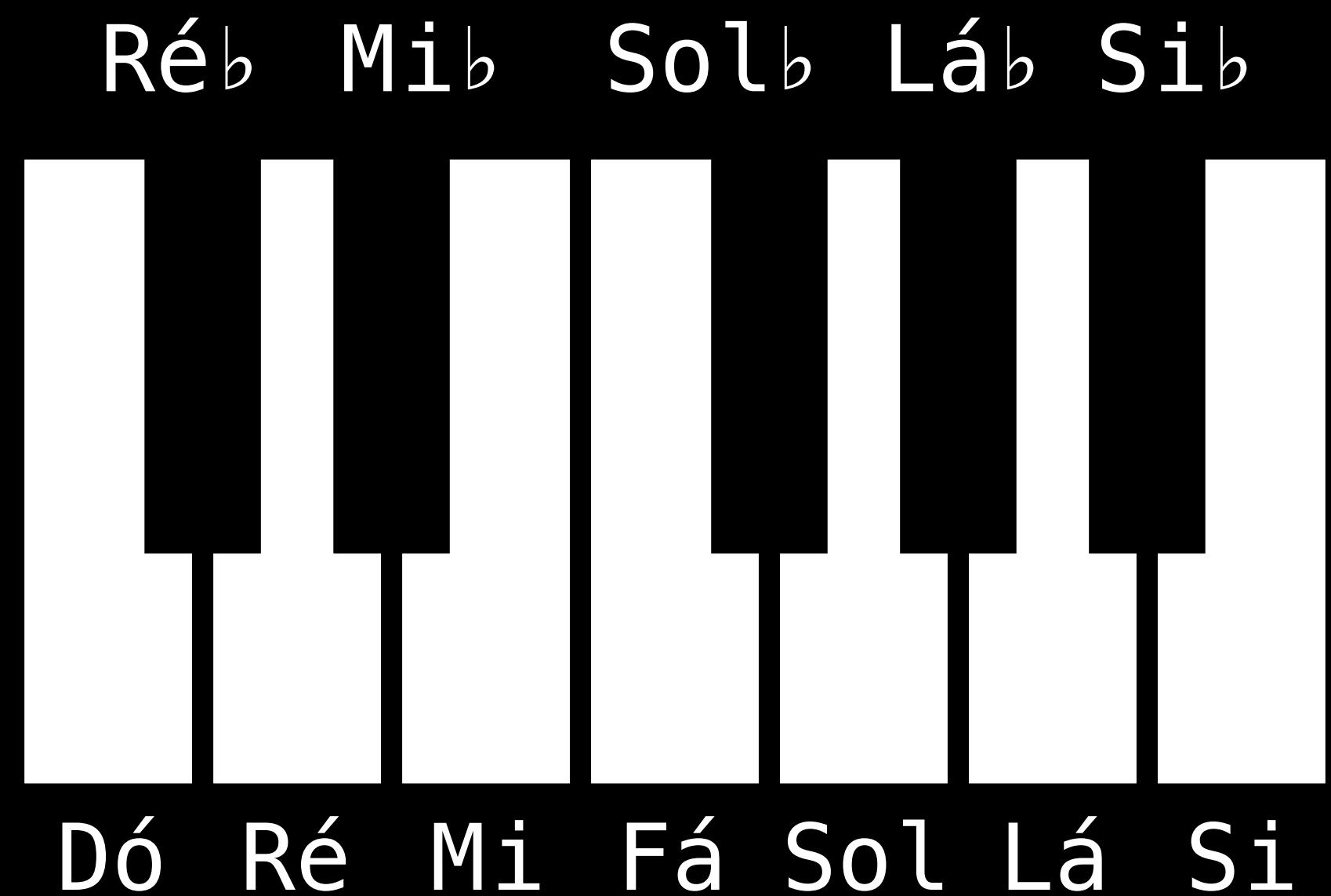
Botão 1
modo Afinador



Botão 2
modo Metrônomo

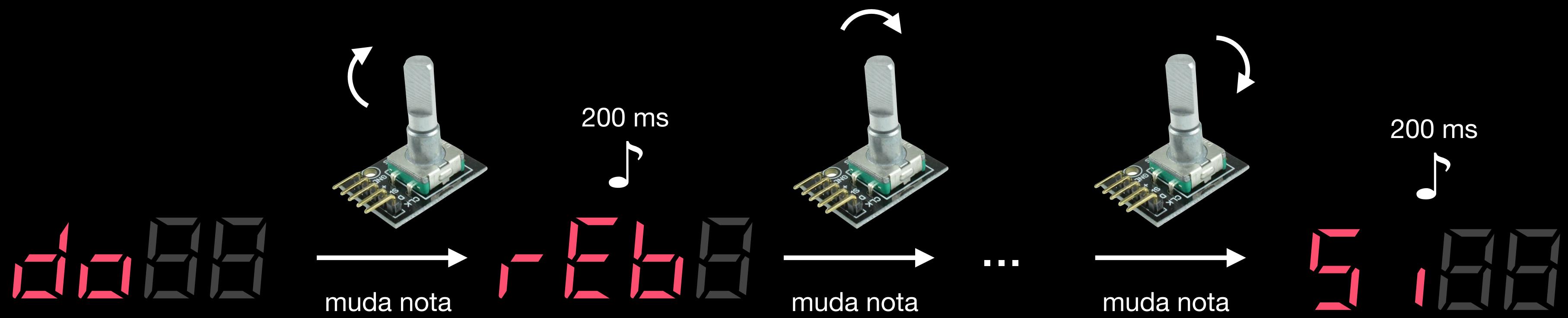


Afinador e Metrônomo



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

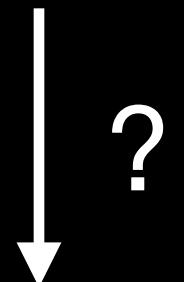
Botão 1



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)

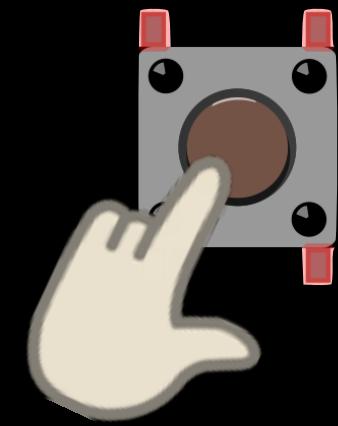
se posição < posição anterior

diminui nota atual (parando em 0)

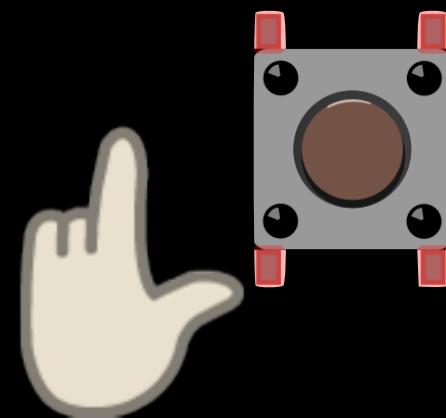


Metrônomo para Marcação de Ritmo

Botão 2



instante1

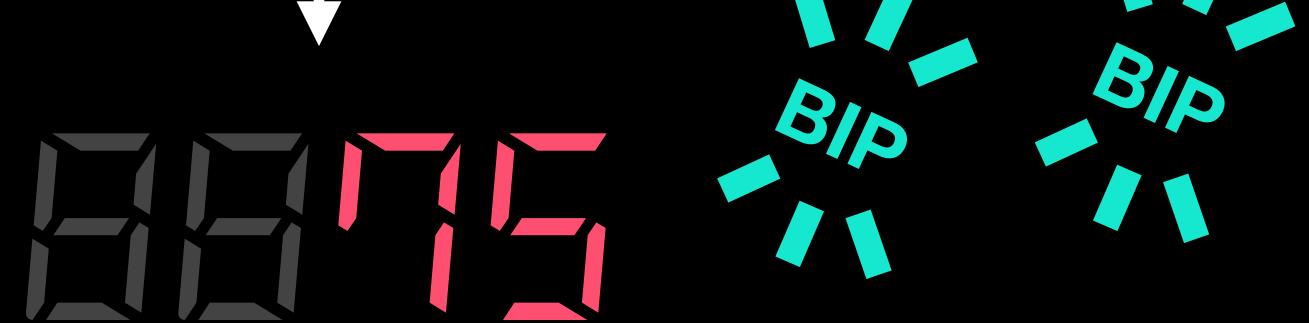


instante2



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

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



(batidas por minuto)

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, registre o instante atual. Ao soltar o Botão 2, obtenha o intervalo de tempo que passou e use isso para exibir a freqüência em batidas por minutos no display.

↪ DICA: veja com cuidado a dica do cálculo no slide anterior.

Também ao soltar o Botão 2, toque um bip periódico. Pare de tocá-lo ao apertar o Botão 1 ou o 2.

↪ DICA: use o Timer1 e a sua função stop(). Lembre-se que esse timer trabalha com tempos em microssegundos.

Aperfeiçoamento



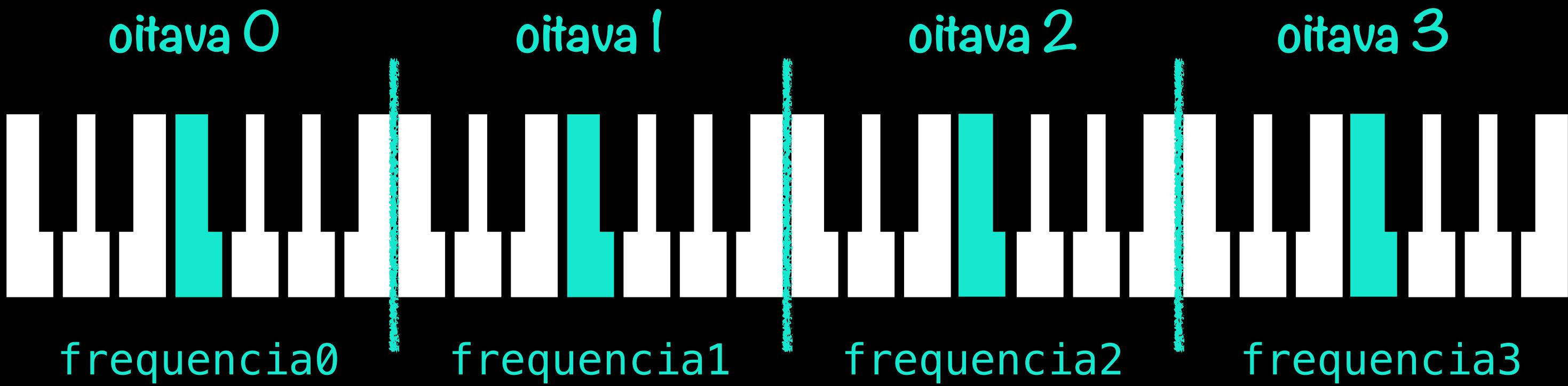
p07b_implementacao.ino

cópia
----->



p07c_aperfeicoamento.ino

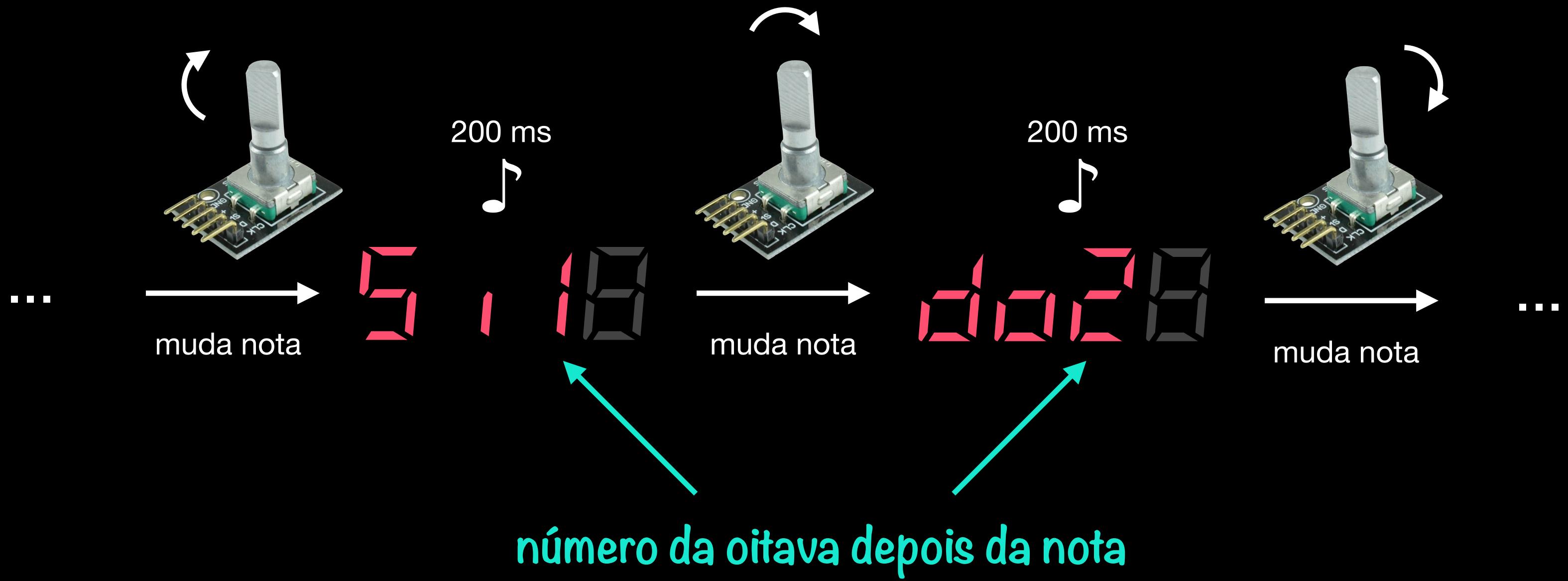
Cópia do Código da Implementação para o Aperfeiçoamento



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



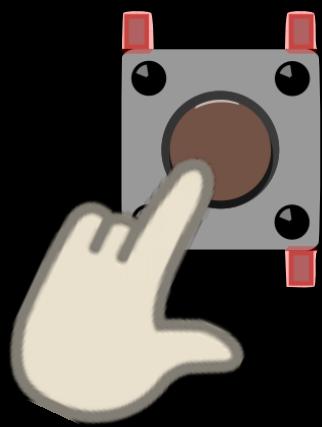
`frequenciaN = frequencia0 * pow(2, N);`



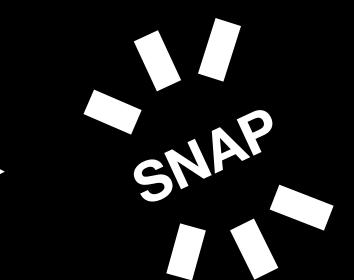
Exibição da Oitava no Display

gravação

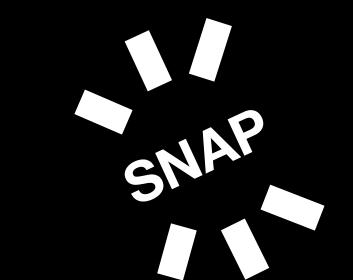
Botão 2



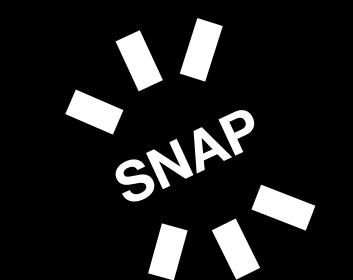
→
0.2s



→
0.7s



→
1.1s

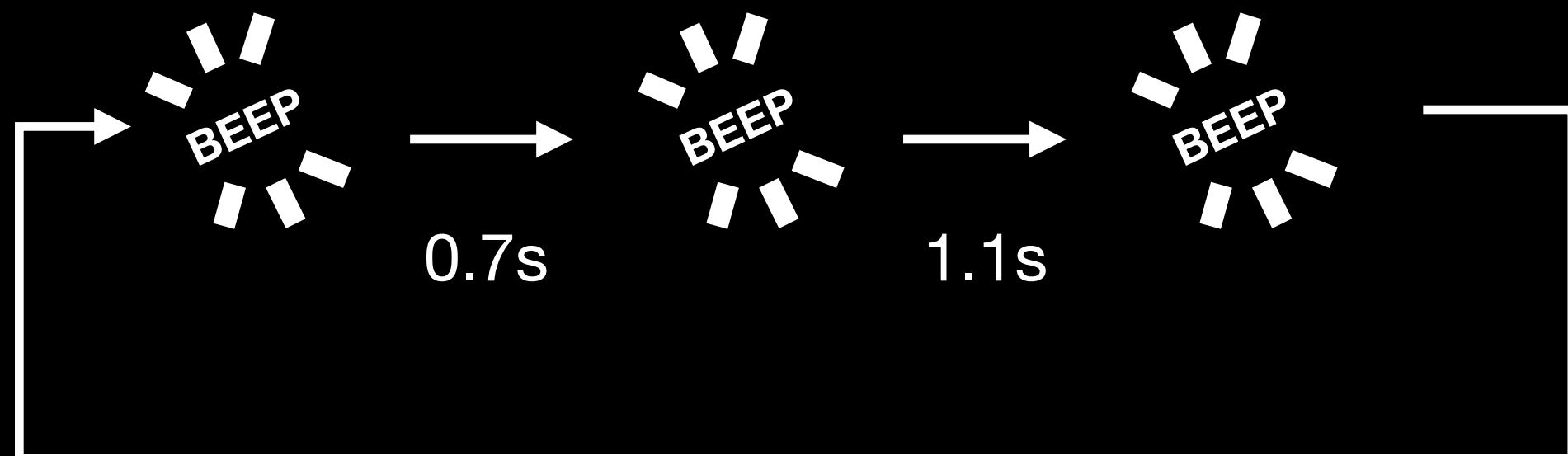


→
0.3s



Botão 2

reprodução



0.2s + 0.3s

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

```
int totalDeEstalos;
```

```
int indiceDoEstaloAtual;
```

Gravação e Reprodução de Padrão de Estalos



Aperfeiçoamento

Ao mexer no encoder no modo metrônomo, **caminhe entre as notas de todas as quatro oitavas**, parando na nota mais baixa ou na mais alta.

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

↪ DICA: pesquise como concatenar uma string com um número inteiro no Arduino.

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

↪ DICA: armazene os tempos em um vetor bem grande, e conte quantos aconteceram.

Também ao soltar o Botão 2, **reproduza essa sequencia de estalos com beeps da campainha (220 HZ e 50ms)**.

↪ DICA: use a Timer1 novamente, mas agora variando o tempo cada vez que chamar a função.

Desafio Extra



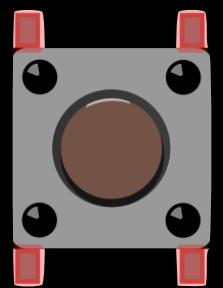
p07c_aperfeicoamento.ino

cópia
----->



p07d_desafio.ino

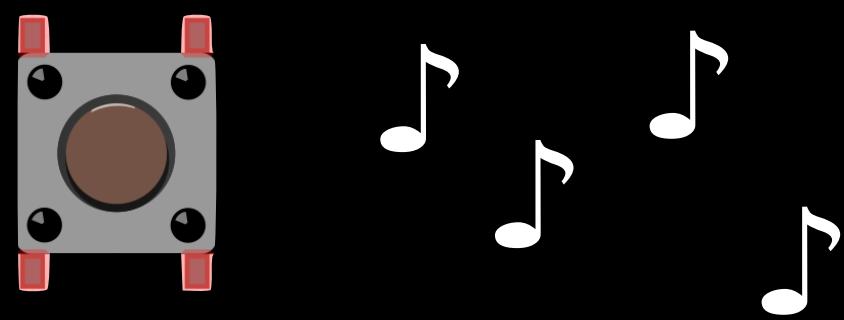
Cópia do Código do Aperfeiçoamento para o Desafio



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

~~Timer1.initialize(...)~~

~~Timer1.attachInterrupt(...)~~

```
unsigned long tempoAtual = millis();
unsigned long intervalo = tempoAtual - tempoDaNotaAnterior;

loop do Arduino:
    se intervalo > intervaloEntreNotas
        toque a nota atual (tone)
        exiba a nota atual no display
        incremente o índice atual
```

Pseudo-Algoritmo para Toque da Música dentro do Loop Principal



Ao apertar o Botão 3, **toque a música indicada**. Use uma duração de 500 ms para cada nota.
↪ DICA: use a função `millis()` dentro do loop do programa (o que permite exibir a cada nota no display).

Desafio Extra



janks.link/micro/projeto07.zip

Material do Projeto 07