



# Projeto 06+07

Controle de Tempo – Prática

Jan K. S. – [janks@puc-rio.br](mailto:janks@puc-rio.br)

ENG1419 – Programação de Microcontroladores

# Testes Iniciais



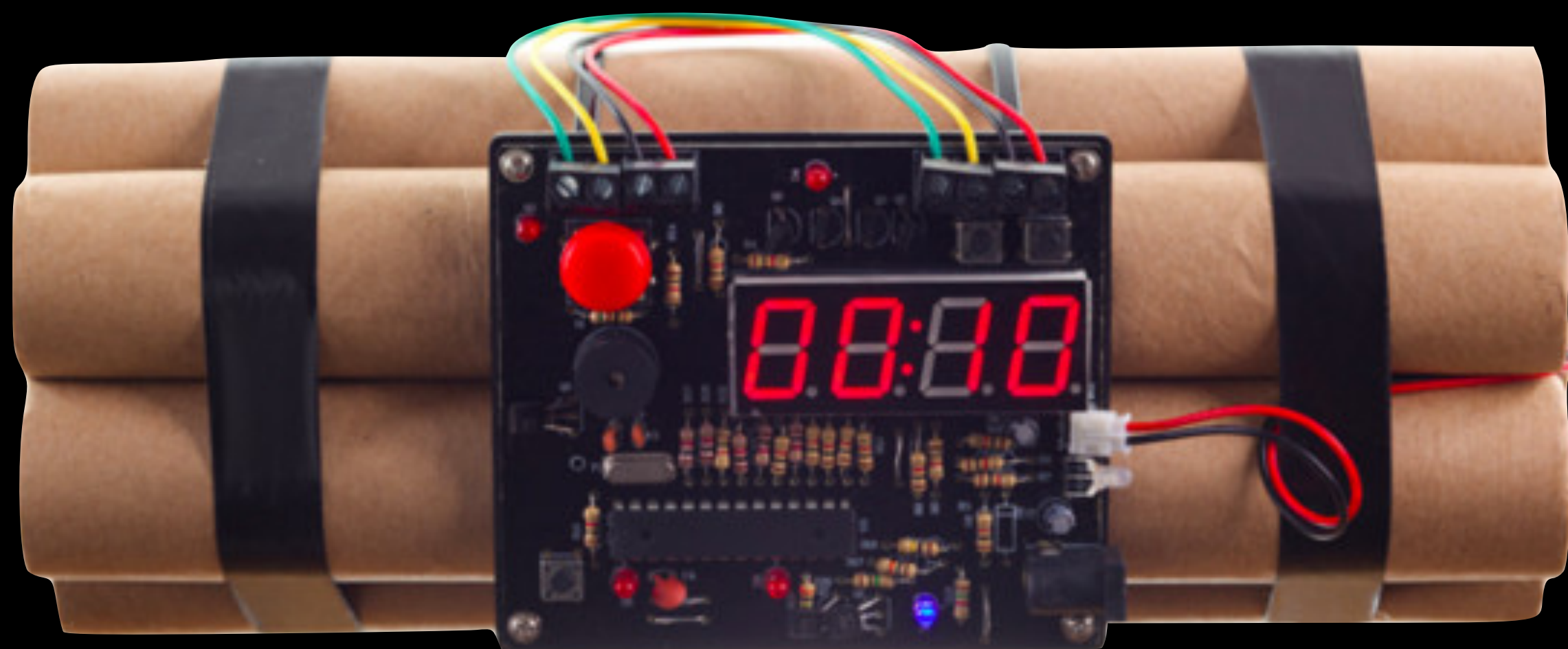


## Testes Iniciais

1. Mantenha o **LED 1 aceso** constantemente.  
↳ DICA: use as funções `pinMode` e `digitalWrite` na `setup`.
2. Ao apertar o **Botão 2**, mude o estado do **LED 2** (entre aceso/apagado).  
↳ DICA: use a `GButton` e uma variável global tipo `bool` (`true/false`) para representar o estado do LED 2.
3. Durante o resto do programa, **exiba no display quantas vezes o Botão 3 foi pressionado**.  
↳ DICA: use a `display.set` e `display.update` na `loop` e uma variável global de contagem.
4. **Imprima a contagem via serial a cada 2 segundos**, sem interferir no display de 7 segmentos.  
↳ DICA: use a função `millis` dentro do `loop`.
5. Ao girar o **encoder no sentido horário**, toque a campainha 1 vez com uma frequência 440Hz durante 50ms. Ao girar no **sentido anti-horário**, toque com 220Hz durante 50ms.  
↳ DICA: não se esqueça de colocar o pino A5 (terra da campainha) no LOW. Compare o valor atual do encoder com o anterior para ver o sentido do giro.

# Implementação

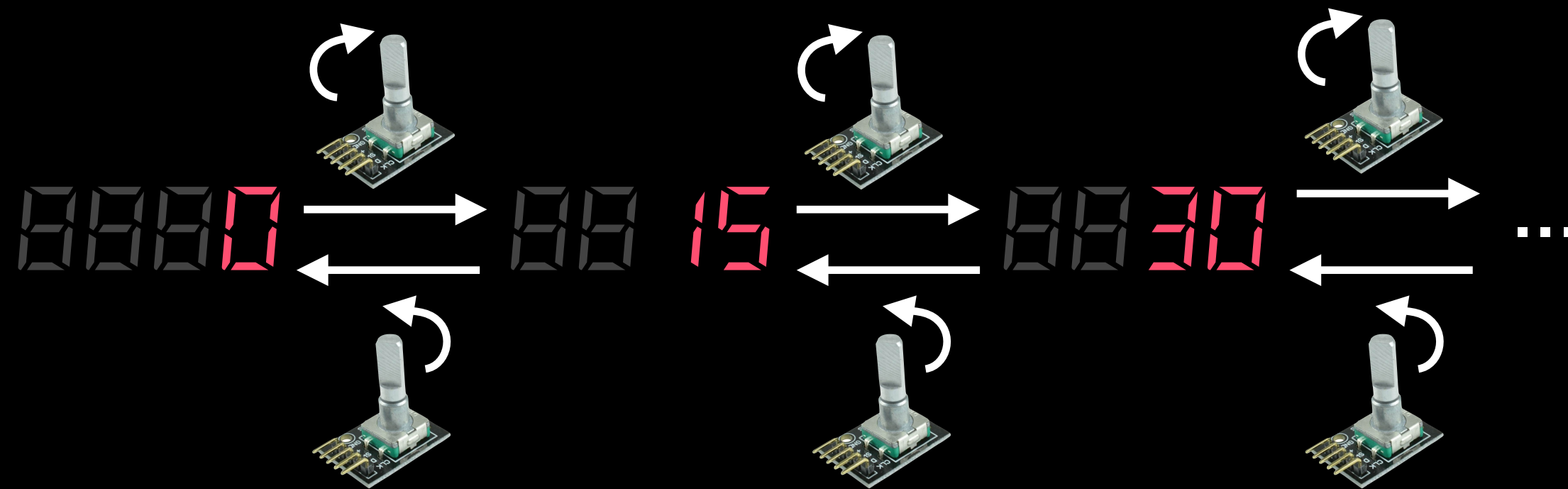




Contagem Regressiva de 7 Segmentos

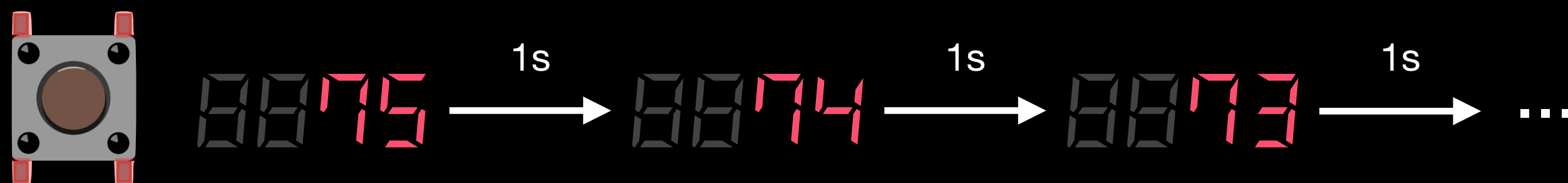
# Encoder

aumenta / diminui tempo (fase de ajuste inicial)



# Botão 1

inicia contagem regressiva



Controle da Contagem Regressiva pelos Botões

88'75 → 01.15

75 segundos



1 minuto e  
15 segundos



OU

envia número 1.15  
para o display

envia número 115  
para o display  
e acende o ponto

Ajuste Final da Exibição do Tempo





## Implementação

1. Crie uma variável global para o tempo da contagem (em segundos). **Exiba continuamente o valor dessa contagem no display** (por enquanto vai ficar só aparecendo zero).  
↳ DICA: use a `display.set` e `display.update` na loop.
2. Adicione 15 nessa contagem **ao girar o encoder no sentido horário** e subtraia 15 **ao girar no sentido anti-horário**. Não permita que o valor fique negativo!
3. A cada 1000 ms, **subtraia 1 da contagem**.
4. Crie uma variável `emAndamento` começando com `false`. **Ao apertar o Botão 1, mude essa variável para `true`**. Em seguida, modifique o item anterior para subtrair 1 da contagem **somente se `emAndamento` for `true`**.
5. Se o valor da contagem chegar a zero e a `emAndamento` for `true`, **mude para `false` e toque a campainha brevemente**.
6. **Mude o formato de exibição de segundos para minutos e segundos** (ex: 00.30 em vez de 30, 01.30 em vez de 90).  
↳ DICA: veja o slide anterior e use divisão e resto da divisão (%).

Aperfeiçoamento



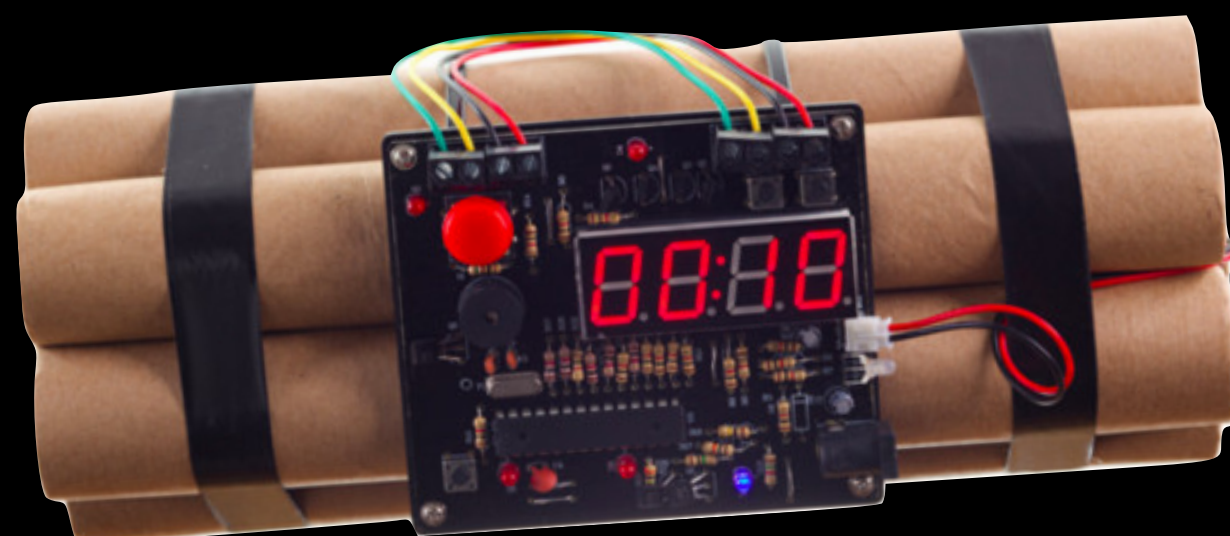
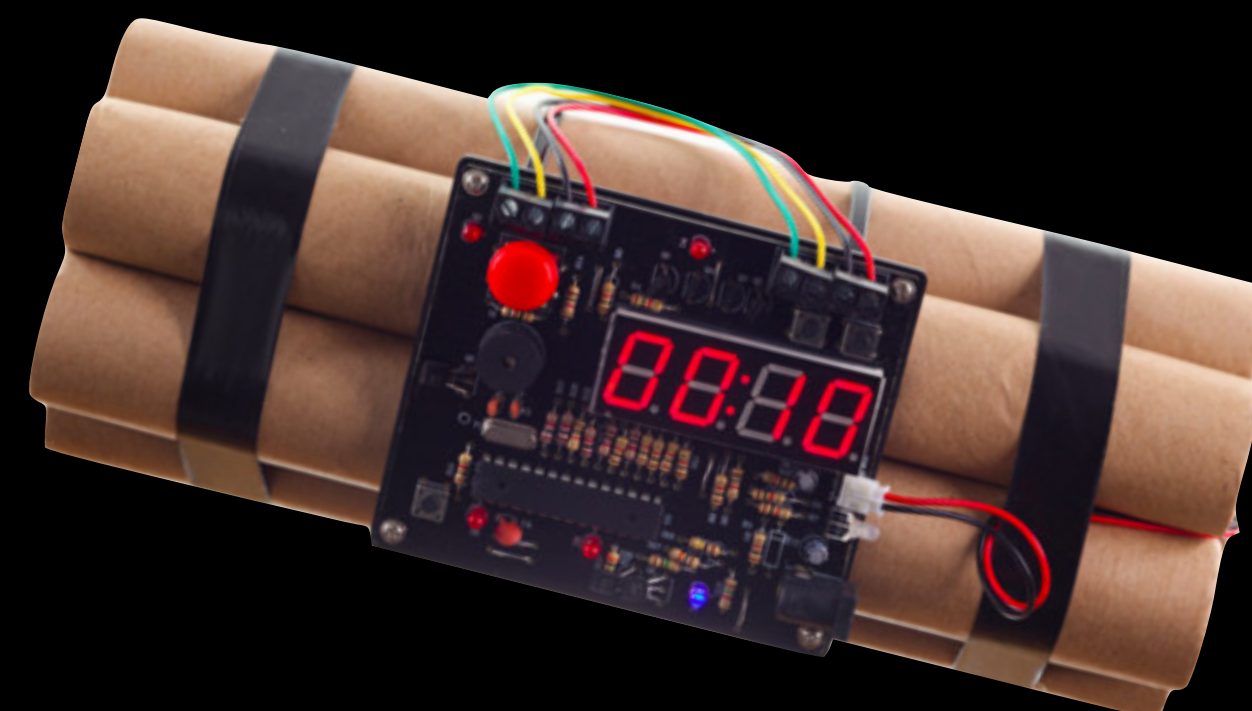
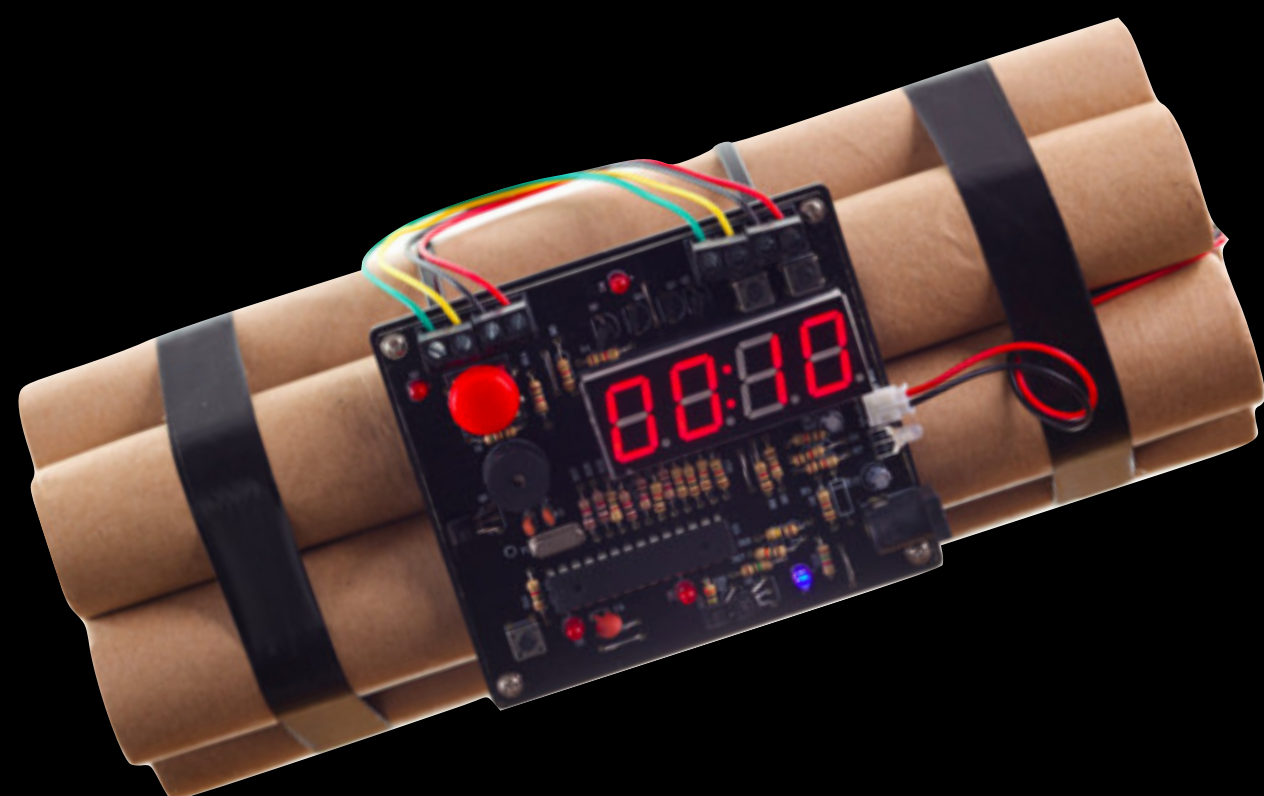
p0607b\_implementacao.ino

cópia  
-----▶



p0607c\_aperfeicoamento.ino

Cópia do Código da Implementação para o Aperfeiçoamento



Múltiplas Contagens

Tempo da contagem atual

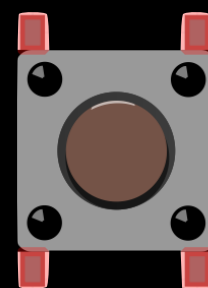
05.33

Indicação da contagem atual



Botão 2

passa para contagem seguinte



Controle de 4 Timers



```
int contagem = 0;  
unsigned long instanteAnterior = 0;  
bool emAndamento = false;
```



```
int contagem[] = {0, 0, 0, 0};  
unsigned long instanteAnterior[] = {0, 0, 0, 0};  
bool emAndamento[] = {false, false, false, false};  
int indiceDaContagemAtual = 0;
```

Conversão de Variáveis para Listas





## Aperfeiçoamento

1. Converta as variáveis da contagem, `emAndamento` e `instanteAnterior` para listas e crie uma variável para o `índice atual` (começando em 0). Corrija essas variáveis ao longo do código, fazendo o acesso nas listas. Teste o programa e veja se ainda funciona com apenas o primeiro contador.
2. Ao apertar o Botão 2, troque o índice para a contagem seguinte. Caso seja a última, volte para a primeira.
3. Acenda o LED relativo à contagem atual que está sendo exibida e apague os demais.
4. Para cada uma das 4 contagens: se estiver em andamento, diminua o valor do tempo a cada segundo. A campainha deve tocar mesmo que a contagem que terminou não esteja sendo exibida.

Desafio Extra



p0607c\_aperfeicoamento.ino

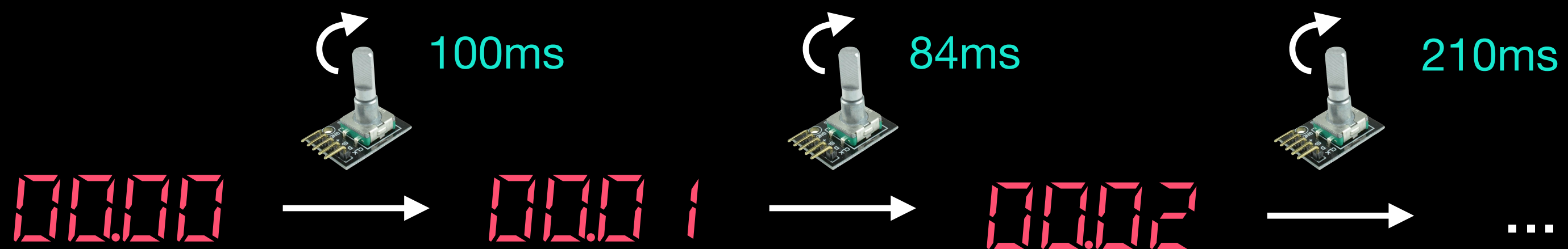
cópia  
-----▶



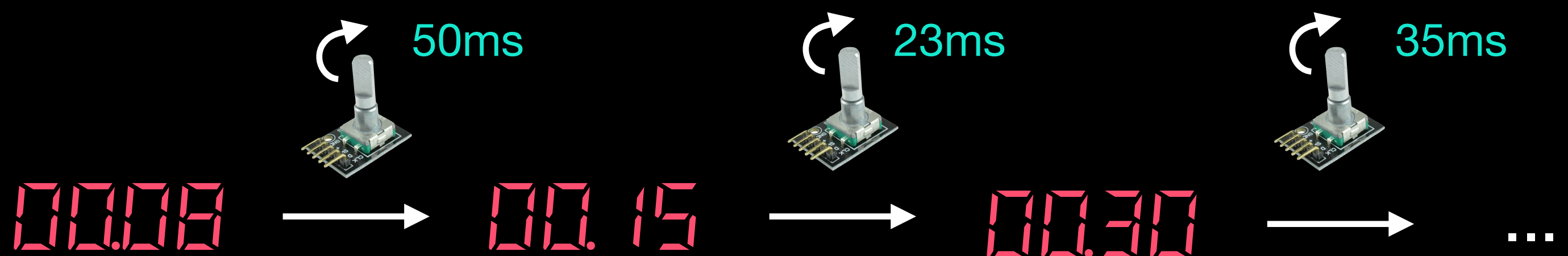
p0607d\_desafio.ino

Cópia do Código do Aperfeiçoamento para o Desafio

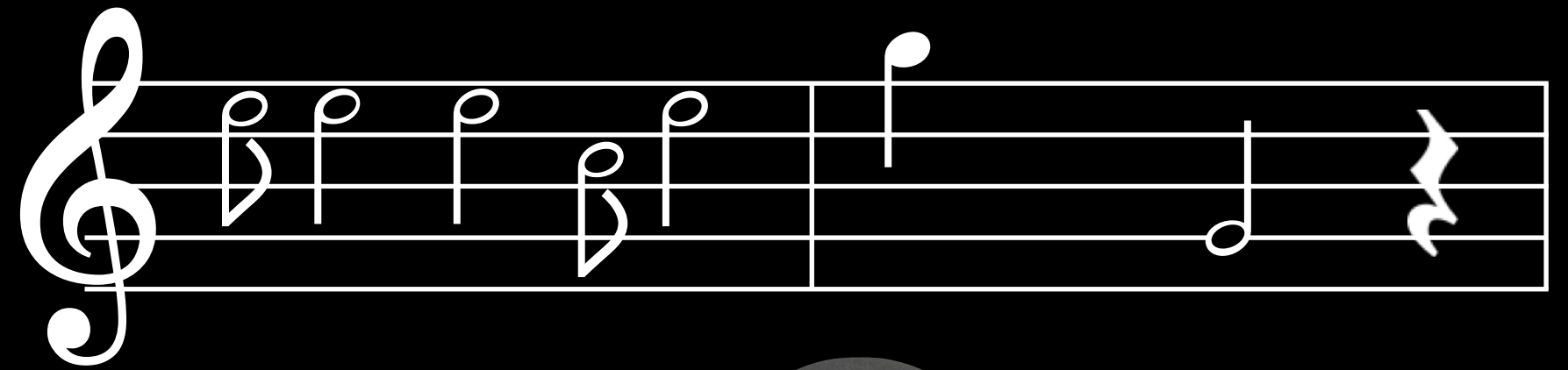
giro lento (aumenta/diminui 1)



giro rápido (vai para próximo múltiplo de 15)



Variação Lenta e Rápida da Contagem



```
int frequenciasDasNotas[7] =  
{659.26, 659.26, 659.26, 523.26, 659.26, 784.00, 392.00};
```

```
int intervalosEntreNotas[7] =  
{100, 200, 200, 100, 200, 400, 200};
```

Música no Final da Contagem



## Desafio Extra

1. Caso o giro do encoder seja lento (mais de 50ms desde o anterior), **adicione/subtraia 1 da contagem atual**. Caso contrário, **mude para o múltiplo de 15 seguinte**.
2. Quanto o tempo terminar, **toque a primeira nota da música**. Em seguida, no loop do Arduino, caso já tenha passado o tempo esperado desde a nota anterior, **toque a próxima nota**. Pare depois de tocar a última nota.



Possíveis Melhorias



## Possíveis Melhorias

Como seriam 3 maneiras diferentes de identificar qual das contagens que acabou de terminar?

Como sinalizar quais contagens estão em andamento?

Como implementar a funcionalidade de pausar / despausar sem perder muita precisão de tempo?



[janks.link/micro/projeto06+07.zip](https://janks.link/micro/projeto06+07.zip)

Material do Projeto 06+07