



# Projeto 06

## Controle de Tempo – Prática

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

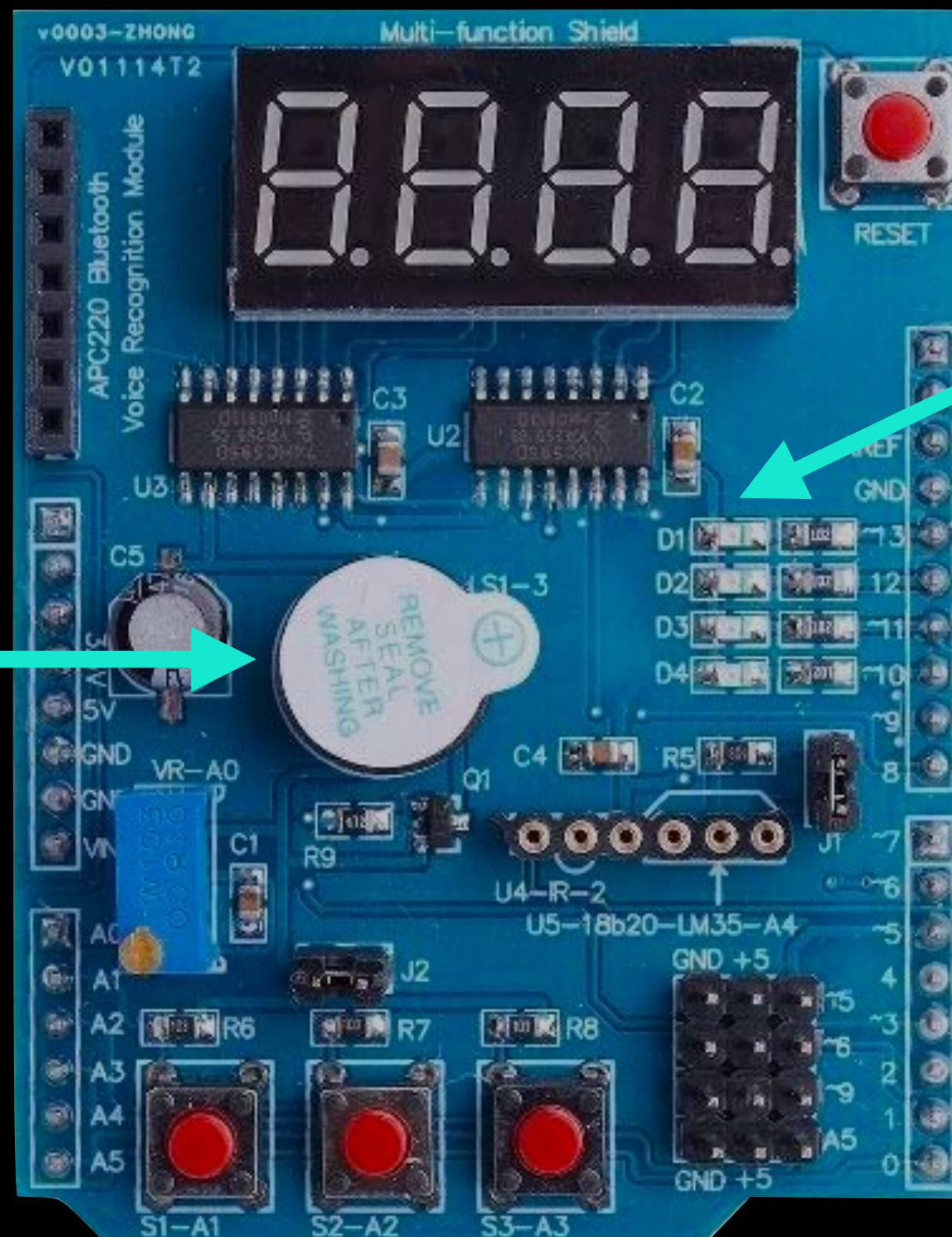
ENG1419 – Programação de Microcontroladores

# Testes Iniciais

pinos 4, 7 e 8

pino 3

pinos 13, 12, 11 e 10



pinos A1, A2 e A3

Pinos Usados pelo Shield Multifunção



## Testes Iniciais

Mantenha o **LED 1 aceso** constantemente.

Se os **Botões 1 e 2 estiverem pressionados ao mesmo tempo**, acenda o LED 2. Caso contrário, ele deve ficar apagado.

↳ DICA: use as funções `pinMode` e `digitalRead`.

**Exiba o número -4.12 no display** por 2 segundos ao iniciar o programa.

↳ DICA: use as funções `set` e `show`.

Durante o resto do programa, **exiba no display quantas vezes o Botão 3 foi pressionado**.

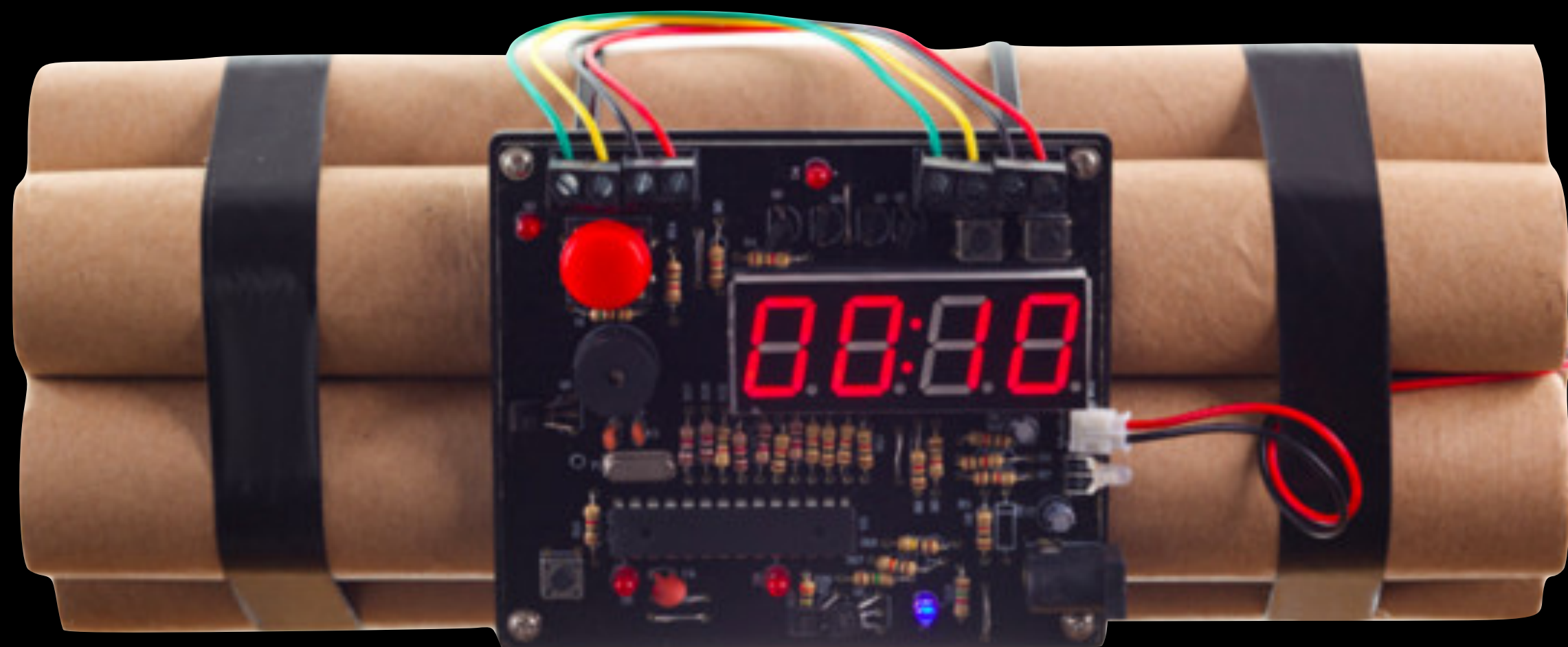
↳ DICA: use a `GFBUTTON` e uma variável global de contagem.

**Imprima a contagem via serial a cada 2 segundos**, sem interferir no display de 7 segmentos.

↳ DICA: use um timer.

# Implementação

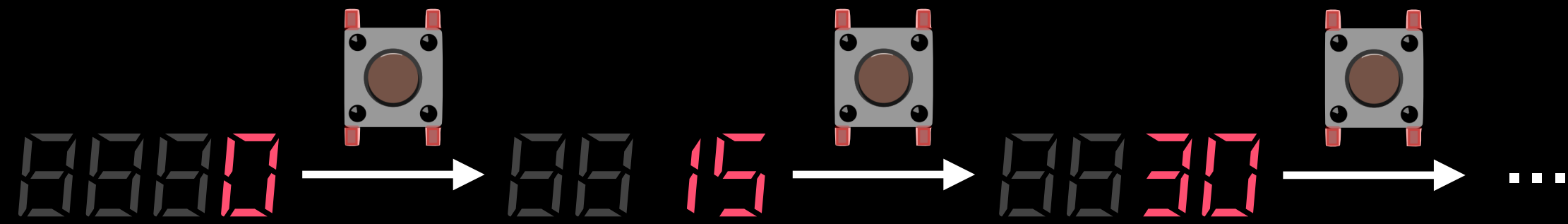




Timer de 7 Segmentos

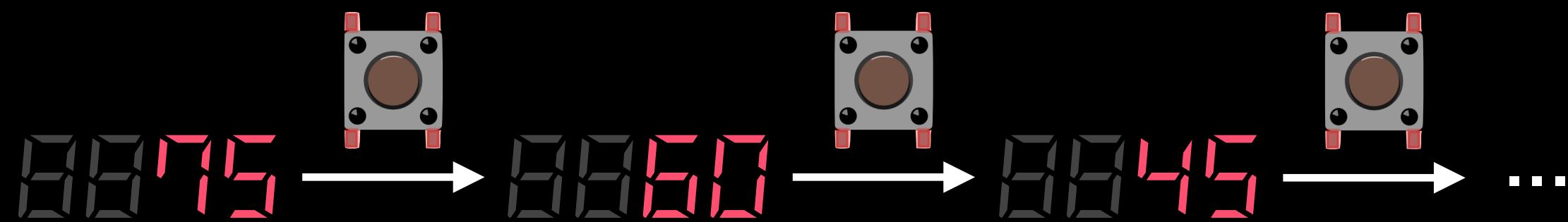
## Botão 1

aumenta tempo



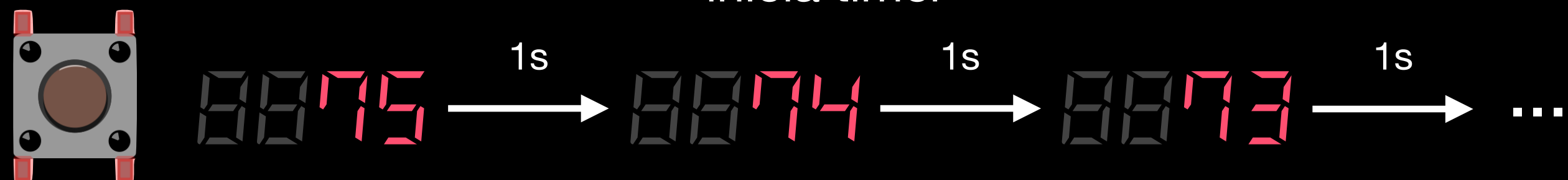
## Botão 2

diminui tempo



## Botão 3

inicia timer



Controle do Timer pelos Botões

88'75



01.15

Ajuste Final da Exibição do Tempo





## Implementação

Exiba o valor do timer (em segundos) no display.

↳ DICA: use uma variável global para o valor.

Aumente o valor do timer ao apertar o Botão 1 e diminua-o ao apertar o Botão 2. Não permita que o valor fique negativo!

↳ DICA: use a GFBUTTON.

Ao apertar o Botão 3, inicie o timer, diminuindo o valor a cada segundo se ele for maior que zero.

↳ DICA: use a biblioteca TimerOne e uma variável global indicando se a contagem está em andamento.

Se o valor chegar a zero e a contagem estiver em andamento, pare a contagem e toque a campainha brevemente.

Mude o formato de exibição de segundos para minutos e segundos (ex: 01.30 em vez de 90).

↳ DICA: use divisão e resto da divisão (%).

Aperfeiçoamento



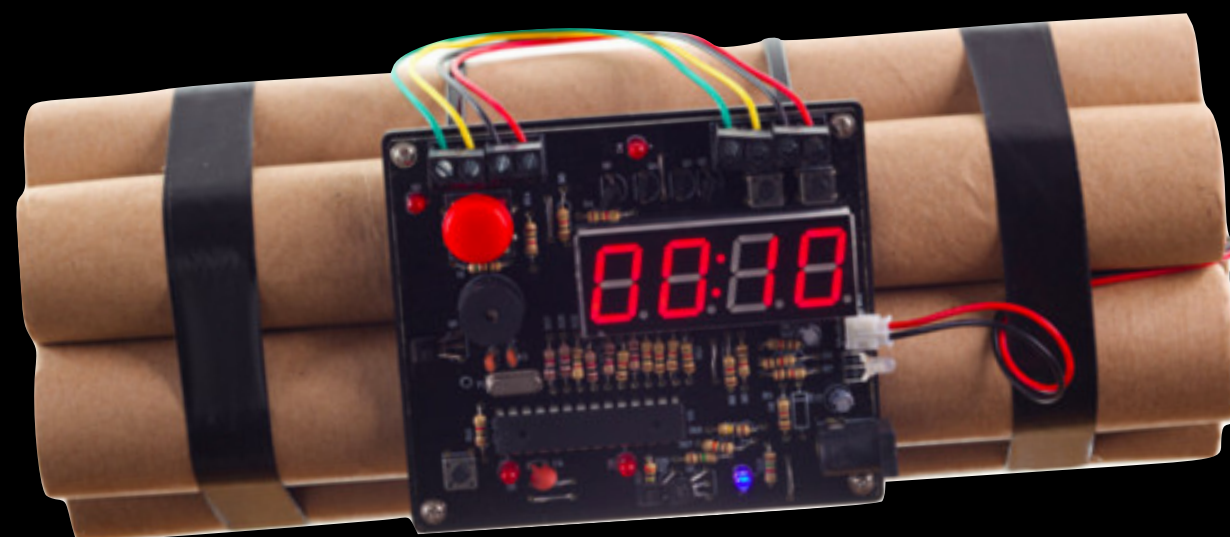
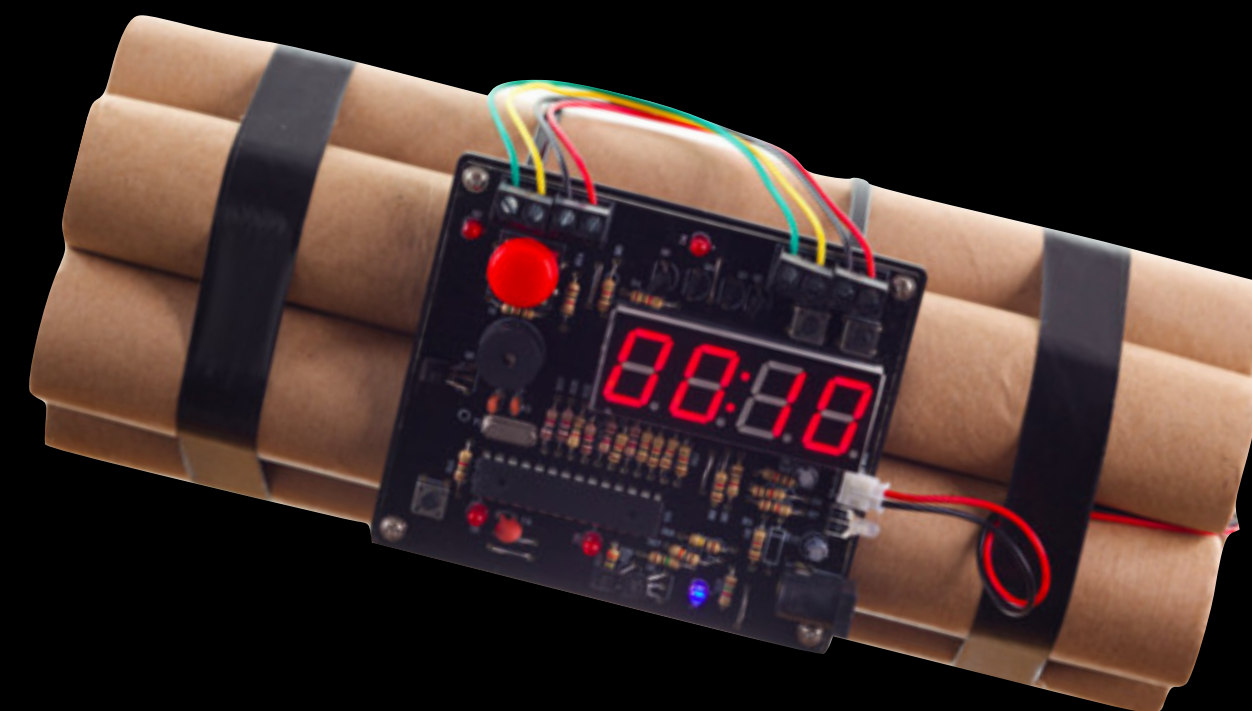
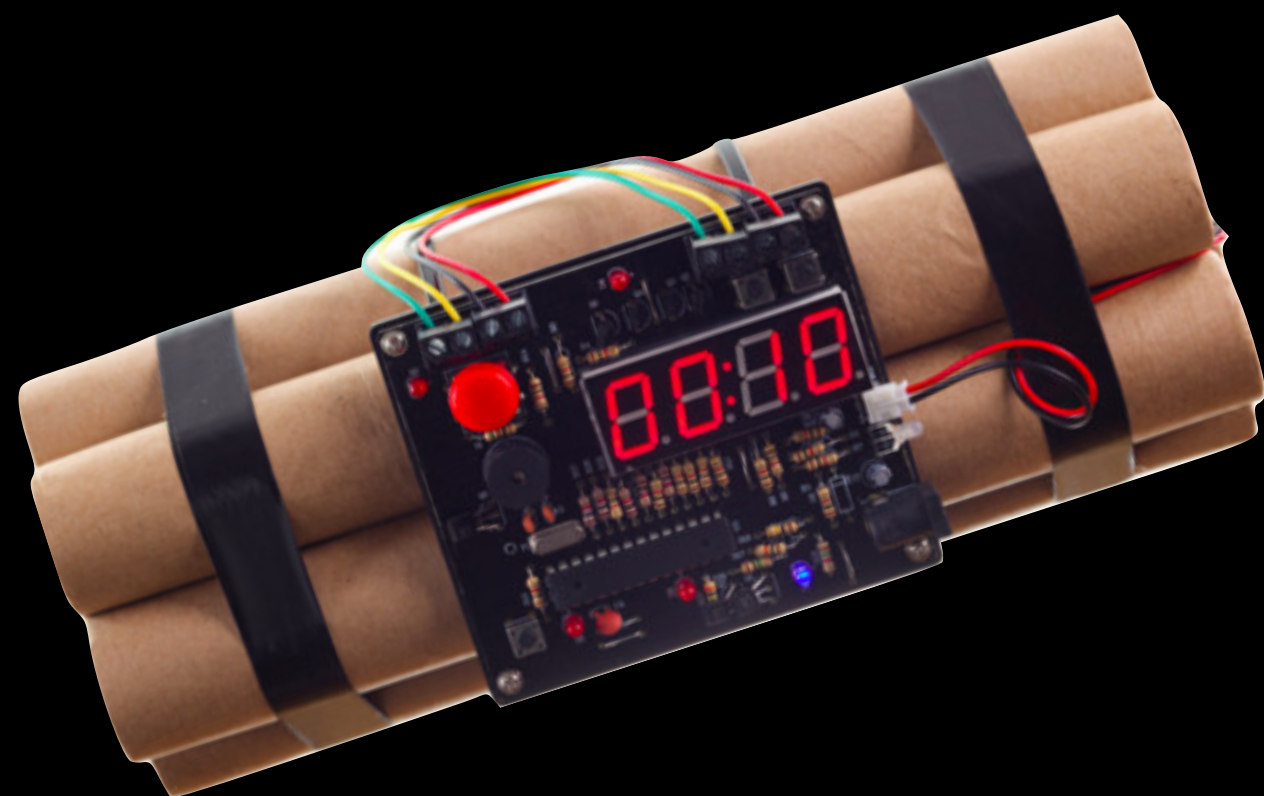
p06b\_implementacao.ino

cópia  
-----▶



p06c\_aperfeicoamento.ino

Cópia do Código da Implementação para o Aperfeiçoamento



Múltiplos Timers

Timer 1  
em andamento

05.33



05.32



05.31



...

Timer 2  
em andamento

07.46



07.45



07.44



...

Timer 3  
parado (em zero)

00.00

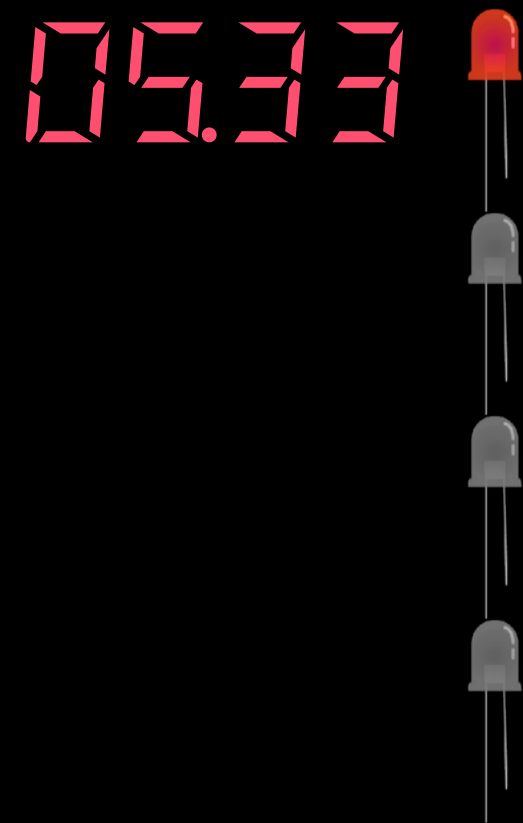
Timer 4  
parado (em ajuste)

0 1.30

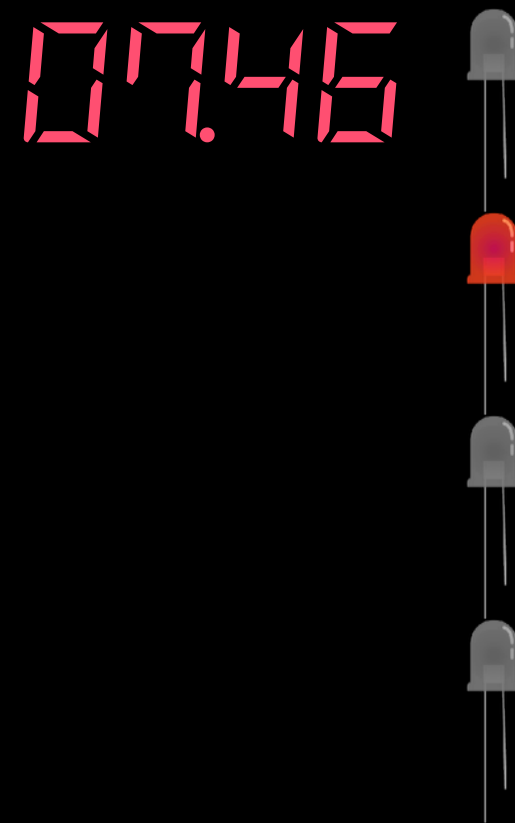
Exemplo de Timers Operando Paralelamente

LEDs indicam qual dos timers está sendo exibido

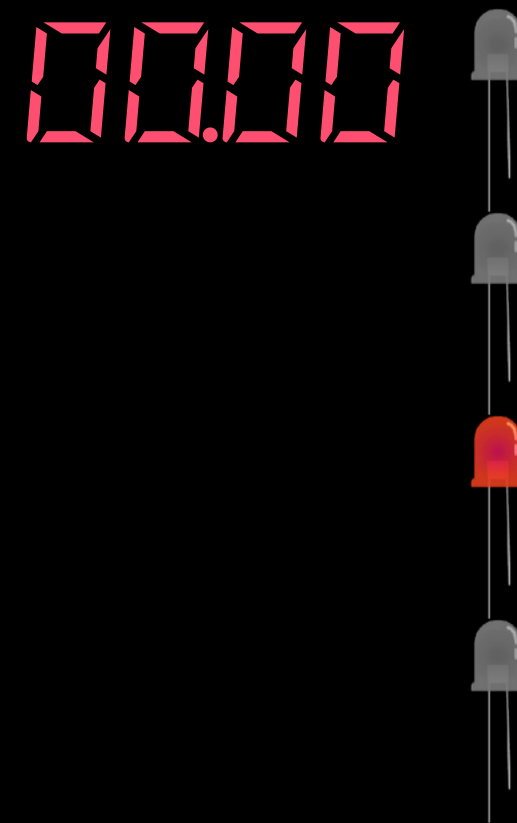
Timer 1



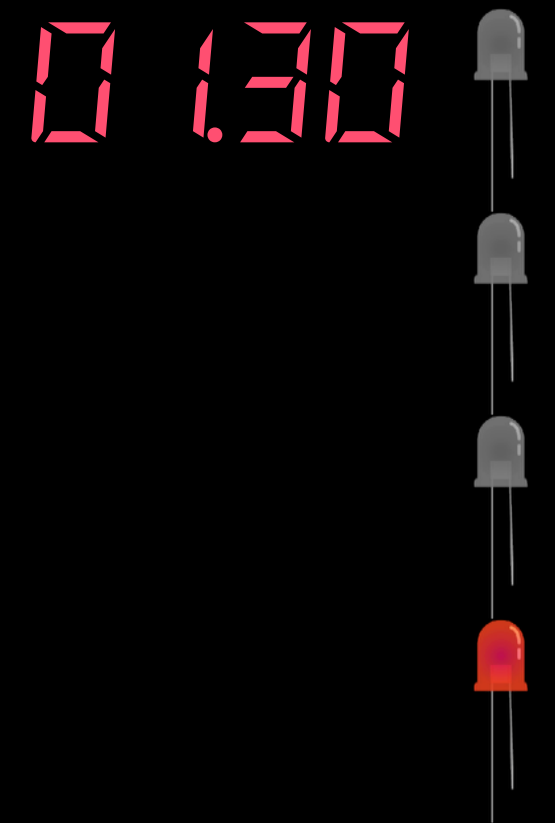
Timer 2



Timer 3

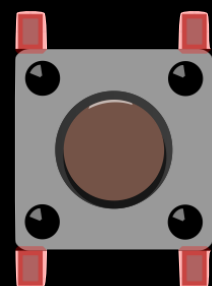


Timer 4



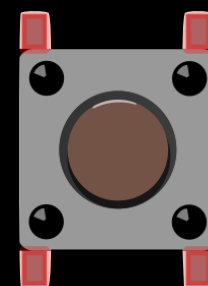
Botão 1

aumenta tempo



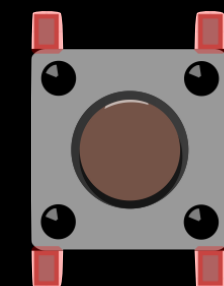
Botão 2

diminui tempo



Botão 3

inicia timer / próximo timer



Controle de Múltiplos Timers



```
int tempo = 0;
```

```
bool emAndamento = false;
```



```
int tempo[] = {0, 0, 0, 0};
```

```
bool emAndamento[] = {false, false, false, false};
```

```
int indiceDoTimerAtual = 0;
```

Conversão de Variáveis para Listas





## Aperfeiçoamento

Converta as variáveis globais para listas. Teste o programa e veja se ainda funciona.

Ao apertar o Botão 3: se o timer atual estiver em andamento ou zerado, troque para o timer seguinte.

Indique qual é o timer que está sendo exibido, acendendo o LED respectivo.

Para cada um dos 4 timers: se estiver em andamento, diminua o valor da contagem a cada segundo.

↪ DICA: modifique a função associada ao Timer1 para lidar com as 4 contagens.

Desafio Extra



p06c\_aperfeicoamento.ino

cópia  
-----▶

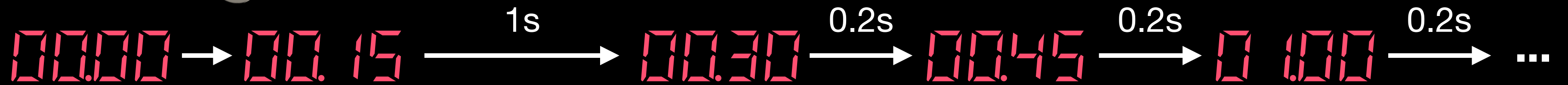
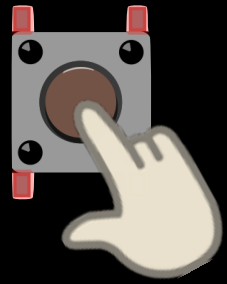


p06d\_desafio.ino

Cópia do Código do Aperfeiçoamento para o Desafio

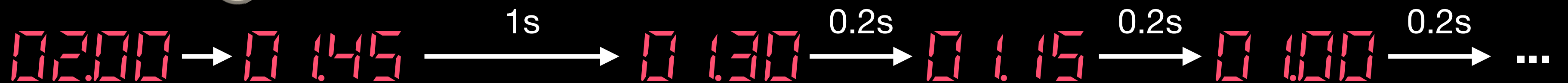
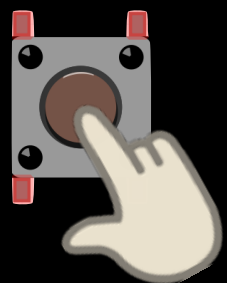
## Botão 1

ao apertar e segurar



## Botão 2

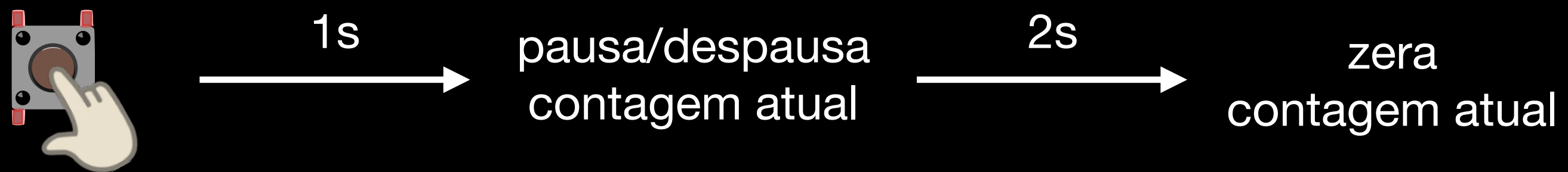
ao apertar e segurar



Melhoria nos Botões 1 e 2

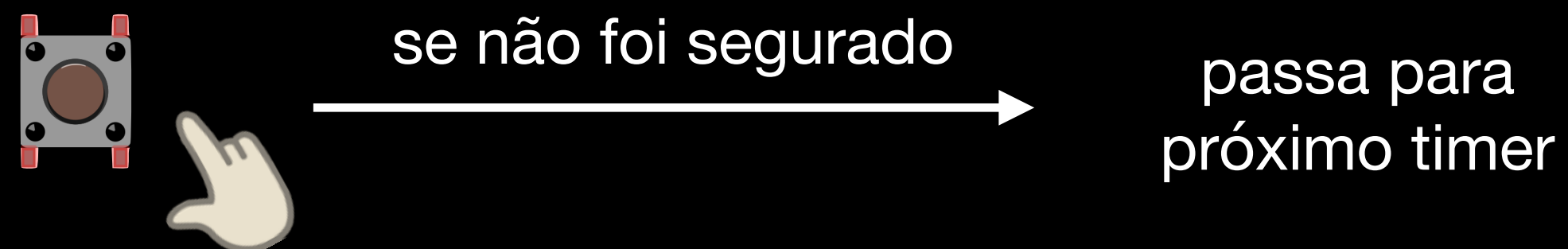
## Botão 3

ao apertar e segurar



## Botão 3

ao soltar

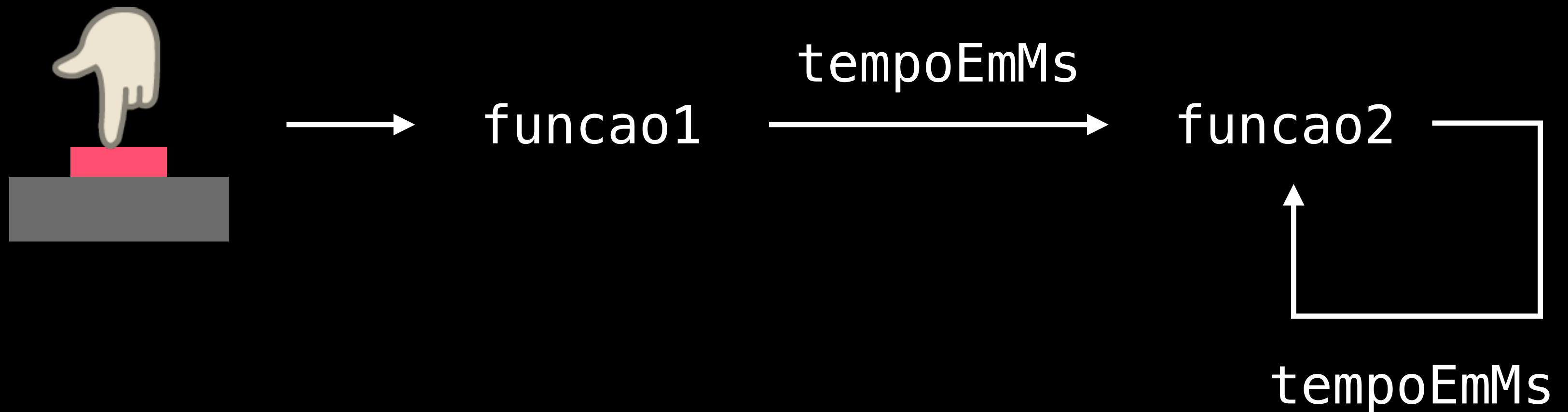


Melhoria no Botão 3

```
botao.setPressHandler(funcao1)
```

```
botao.setHoldHandler(funcao2)
```

```
botao.setHoldTime(tempoEmMs)
```



Funções para Evento de Botão Segurado



## Desafio Extra

Ao segurar o Botão 1 ou o Botão 2 por mais de 1 segundo, **aumente/diminua 15s do tempo a cada 200 ms** enquanto o botão estiver sendo segurado.

↳ DICA: use a função `setHoldHandler` e `setHoldTime`.

Mude o comportamento anterior do Botão 3. Se ele estiver for segurado por 1 segundo, **pause ou despause imediatamente o timer atual**. Se ele se mantiver segurado por mais 2 segundos, **zere a contagem imediatamente, sem tocar a campainha**. Ao soltar o botão, **mude para o próximo timer** caso o botão não tenha sido segurado por mais de 1 segundo.

↳ DICA: não use mais a função `setPressHandler`.





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

Material do Projeto 06