



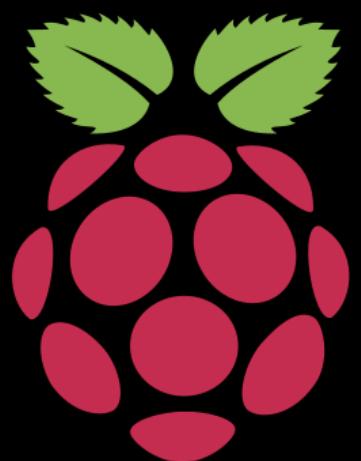
Projeto 07

Controle Sonoro – Teoria

Jan K. S. – janks@puc-rio.br

ENG1419 – Programação de Microcontroladores

"Software"



sleep(1)

```
timer = Timer(2, funcao)  
timer.start()
```

datetime.now()

delay(1000)

```
Timer1.initialize(2000000);  
Timer1.attachInterrupt(funcao);
```

millis()

The screenshot shows a web browser window with the following details:

- Address Bar:** www.arduino.cc/reference/en/language/functions/time/millis
- Title Bar:** Arduino Reference
- Page Content:**
 - Path:** Reference > Language > Functions > Time > Millis
 - Section Header:** millis()
 - Category:** [Time]
 - Description:** Returns the number of milliseconds passed since the Arduino board began running the current program. This number will overflow (go back to zero), after approximately 50 days.
 - Syntax:** time = millis()

Início do Programa



1.5s

`millis()` → 1500



20s

`millis()` → 21500



150s

`millis()` → 171500

Exemplo de Contagem de Milissegundos com a Millis

ion index

Arduino reference lists these datatypes:

Datatype	RAM usage
void keyword	N/A
boolean	1 byte
char	1 byte
unsigned char	1 byte
int	2 byte
unsigned int	2 byte
word	2 byte
long	4 byte
unsigned long	4 byte
float	4 byte

máximo = 32767

máximo = 65535

máximo = 4294967295



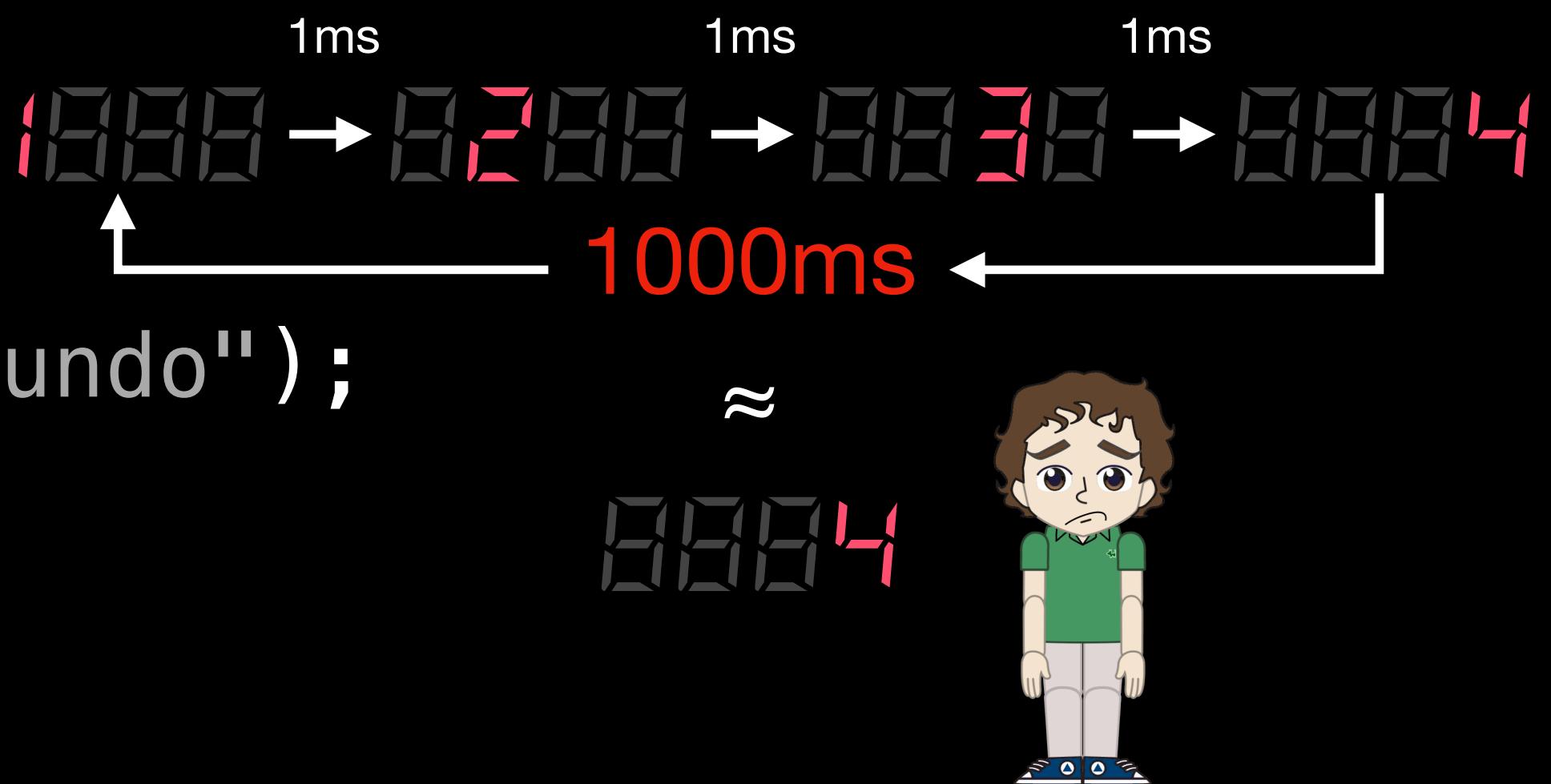
A cartoon illustration of a young boy with brown hair, wearing a green polo shirt. He is pointing his right index finger upwards towards the text "máximo = 4294967295". A black curved line connects the boy's arm to the text.

Usamos unsigned long para poder contar tempo sem esbarrar no limite de 2 bytes.

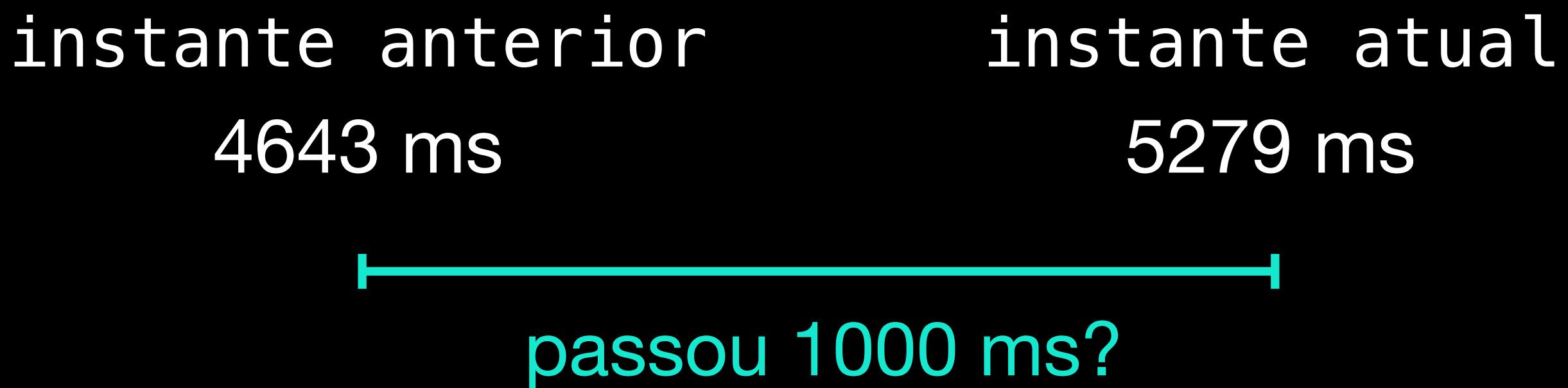
```
#include <ShiftDisplay.h>
```

```
ShiftDisplay display(4, 7, 8, COMMON_ANODE, 4, true);
void setup () {
    Serial.begin(9600);
    display.set(1234);
}
```

```
void loop () {
    display.update();
    delay(1000);
    Serial.println("+1 segundo");
}
```



Relembrando o Problema com a Função Delay



loop:

se tiver passado 1 segundo **desde o instante anterior**:
imprime texto na serial
salva instante de tempo atual

```
#include <ShiftDisplay.h>

ShiftDisplay display(4, 7, 8, COMMON_ANODE, 4, true);
unsigned long instanteAnterior = 0;

void setup () {
    Serial.begin(9600);
    display.set(1234);
}

void loop () {
    display.update();

    unsigned long instanteAtual = millis();
    if (instanteAtual > instanteAnterior + 1000) {
        Serial.println("+1 segundo");
        instanteAnterior = instanteAtual;
    }
}
```

```
unsigned long instanteDaContagem1 = 0;
unsigned long instanteDaContagem2 = 0;
unsigned long instanteDaContagem3 = 0;

void loop () {
    unsigned long instanteAtual = millis();
    if (instanteAtual > instanteDaContagem1 + 1000) {
        ...
    }

    if (instanteAtual > instanteDaContagem2 + 200) {
        ...
    }

    if (instanteAtual > instanteDaContagem3 + 8000) {
        ...
    }

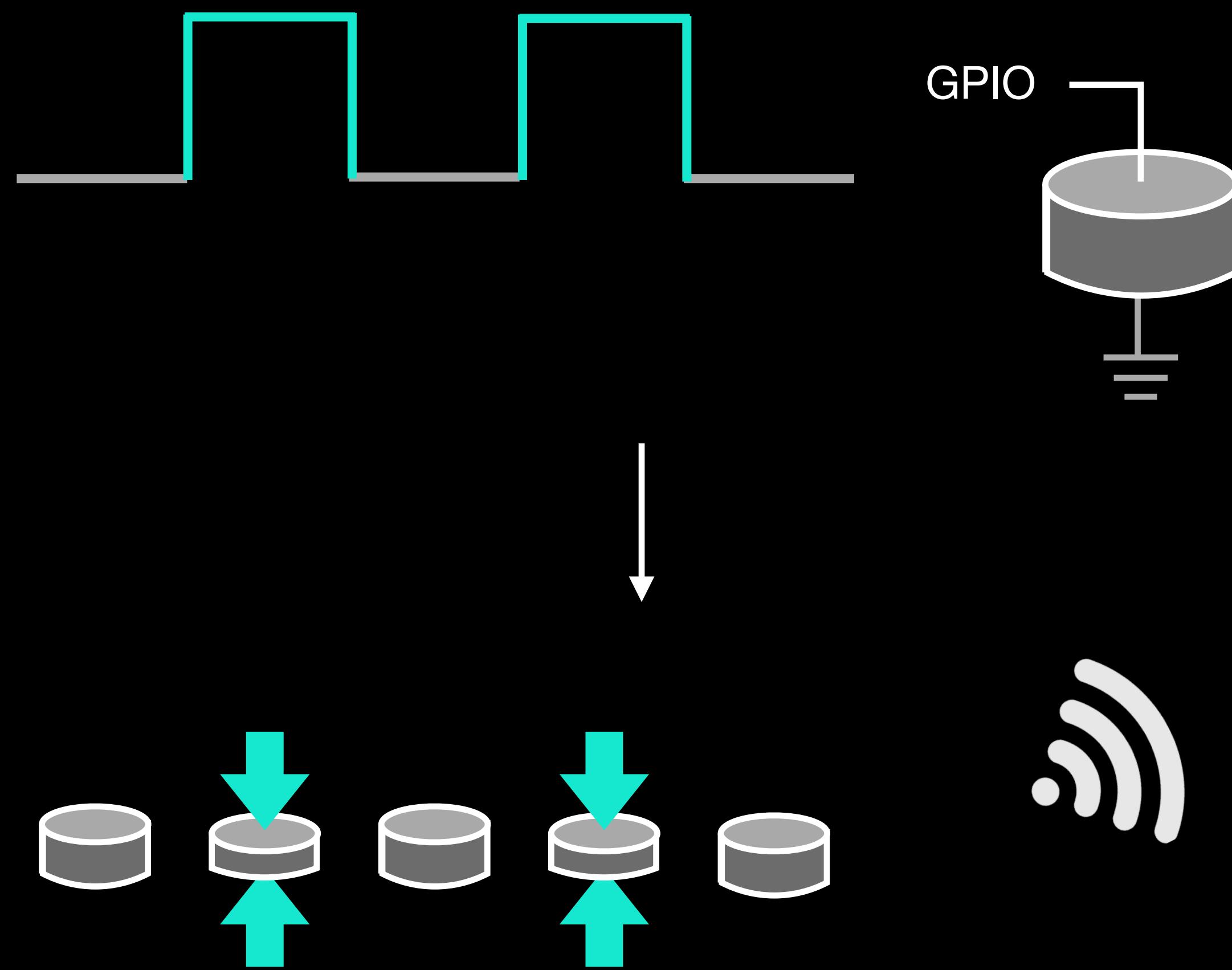
    ...
}
```

Várias Contagens de Tempo com a Função Millis

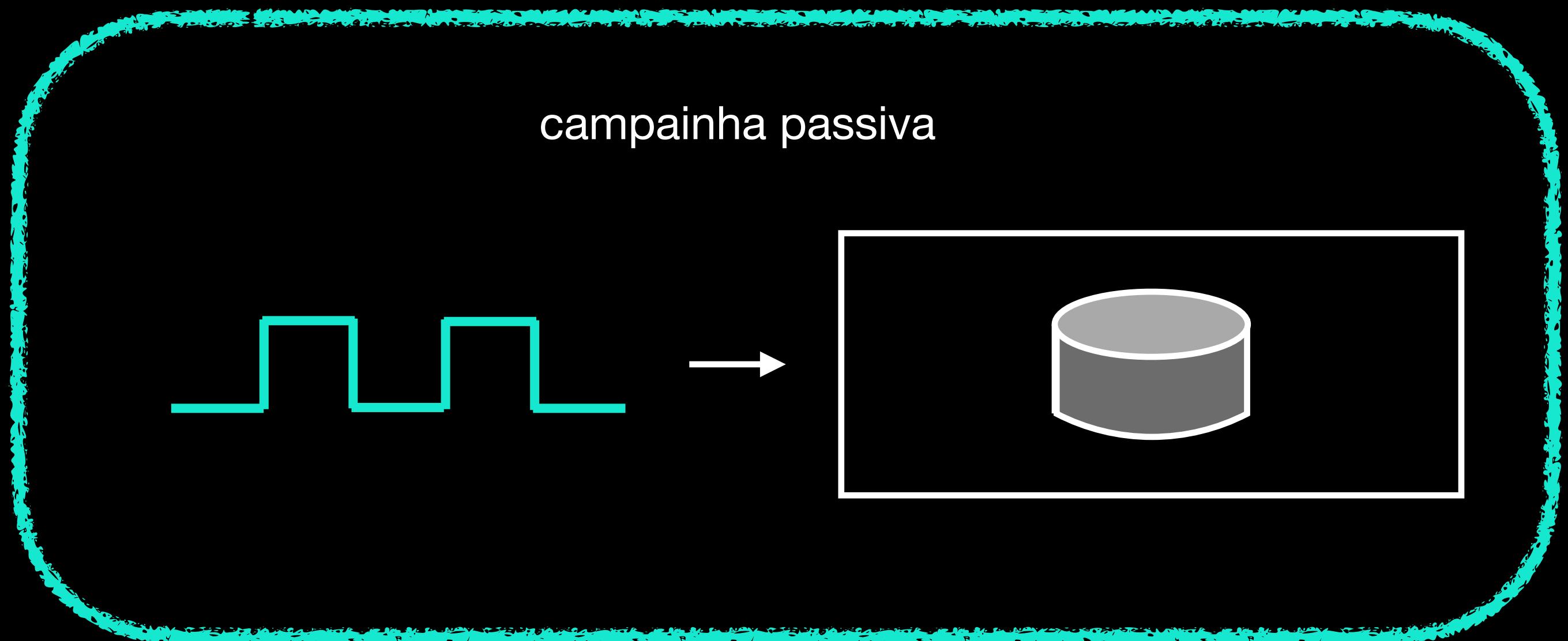
Hardware



Campainha (Buzzer)



Geração de Som por Pulso

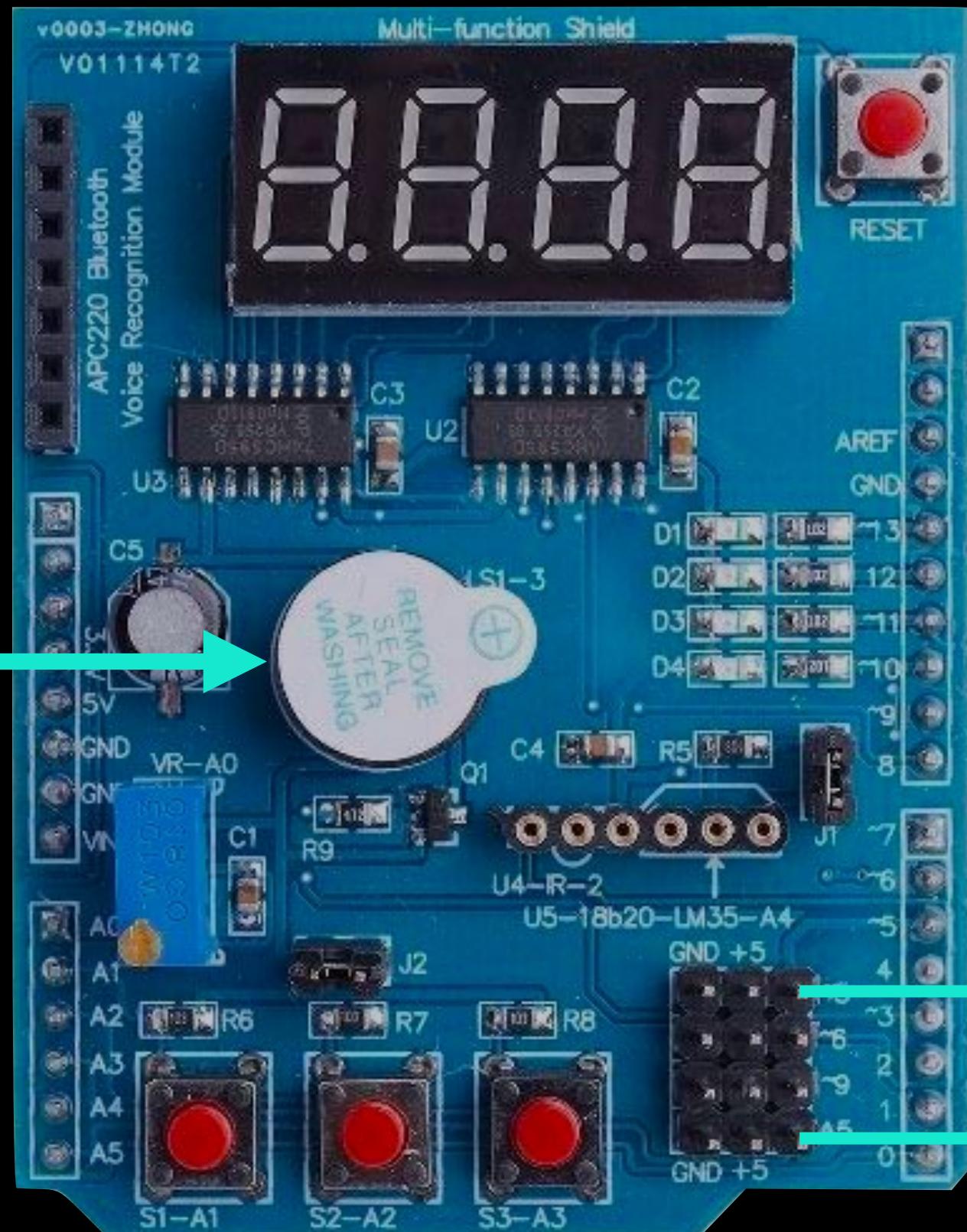


campainha ativa



Campainhas Ativas e Passivas

campainha
ativa



campainha
passiva

5

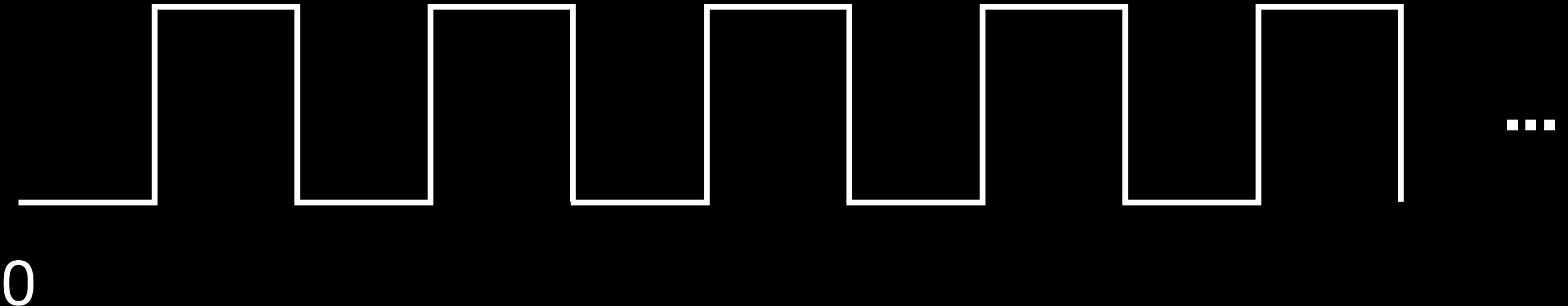
A5

LOW

Campainha Passiva Conectada a Duas Portas de Uso Geral

pino →
tone(5, 440);
freqüência em Hz

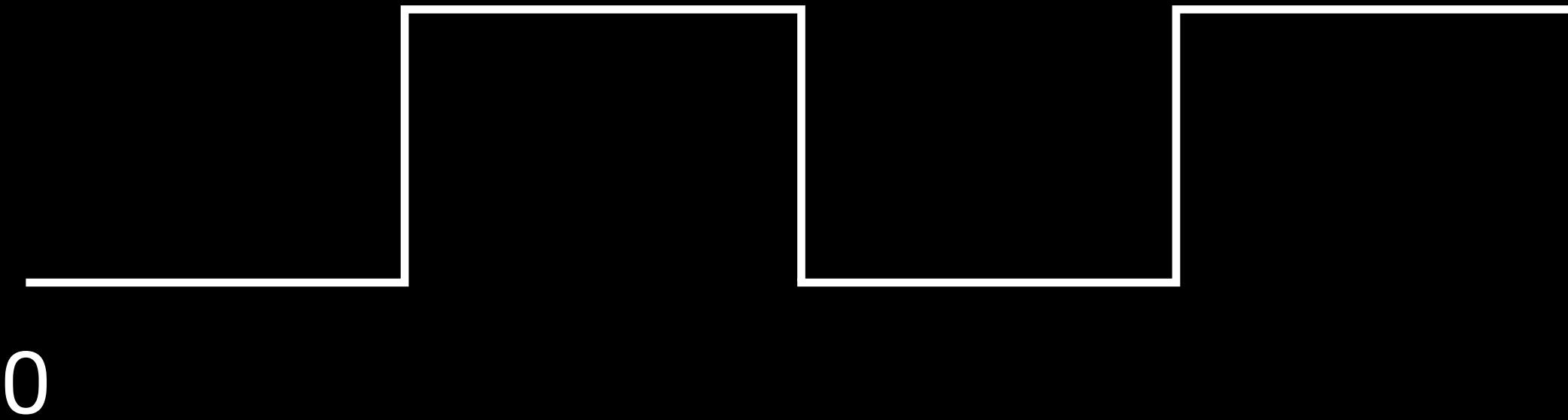
som agudo



noTone(5)
(silêncio)

som grave

duração em ms (opcional)
tone(5, 220, 500);



Frequencia do Sinal para a Campainha

```
int terra = A5;
int campainha = 5;

void setup () {
    pinMode(terra, OUTPUT);
    digitalWrite(terra, LOW);

    pinMode(campainha, OUTPUT);

    // sinal de 220 Hz durante 500 milissegundos
    tone(campainha, 220.0, 500);
}
```

```
int terra = A5;  
int campainha = 5;  
  
void setup () {  
    pinMode(terra, OUTPUT);  
    digitalWrite(terra, LOW);  
  
    pinMode(campainha, OUTPUT);  
  
    tone(campainha, 220.0, 500);  
    tone(campainha, 440.0, 500);  
}
```

A função tone **não trava**
a execução do programa

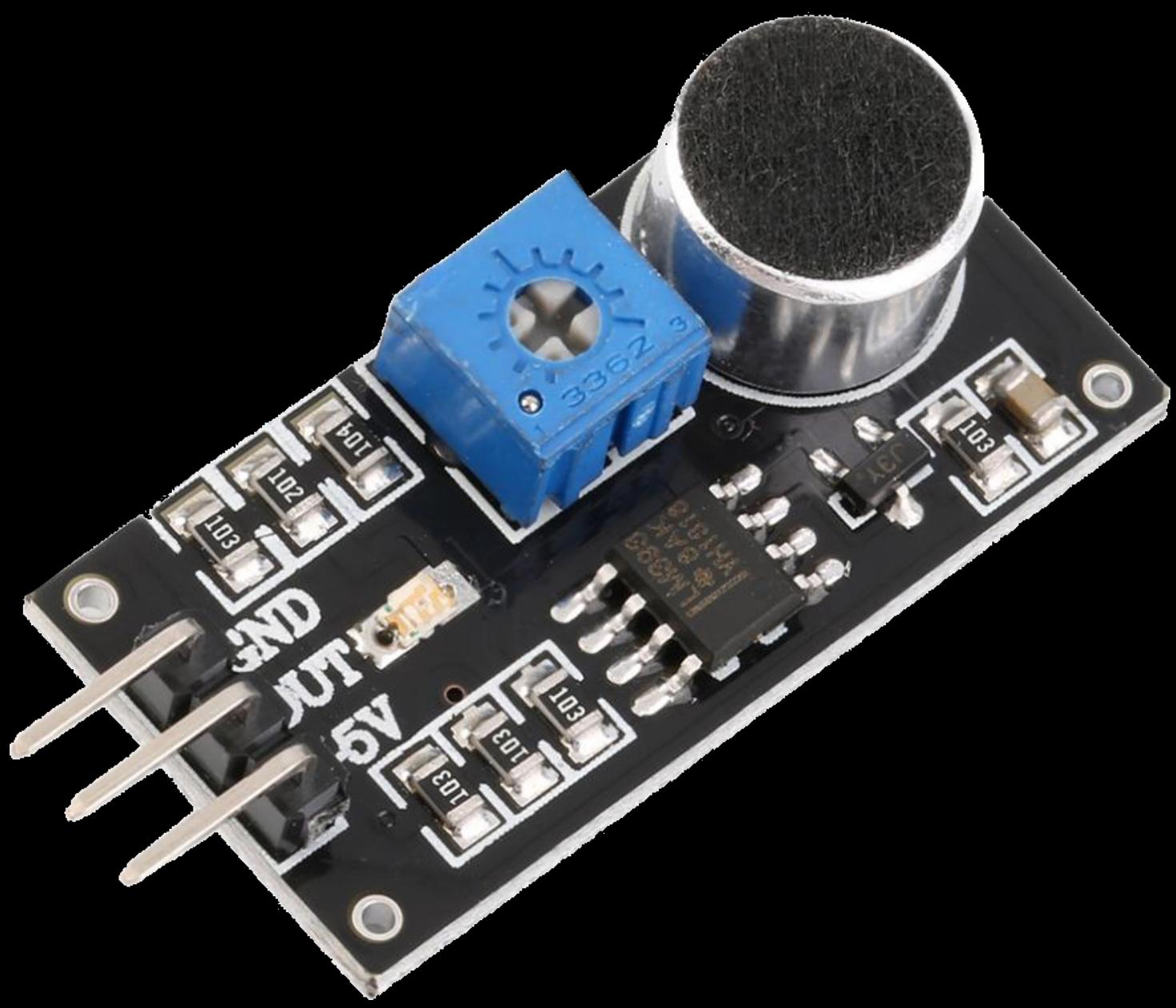


```
int terra = A5;
int campainha = 5;

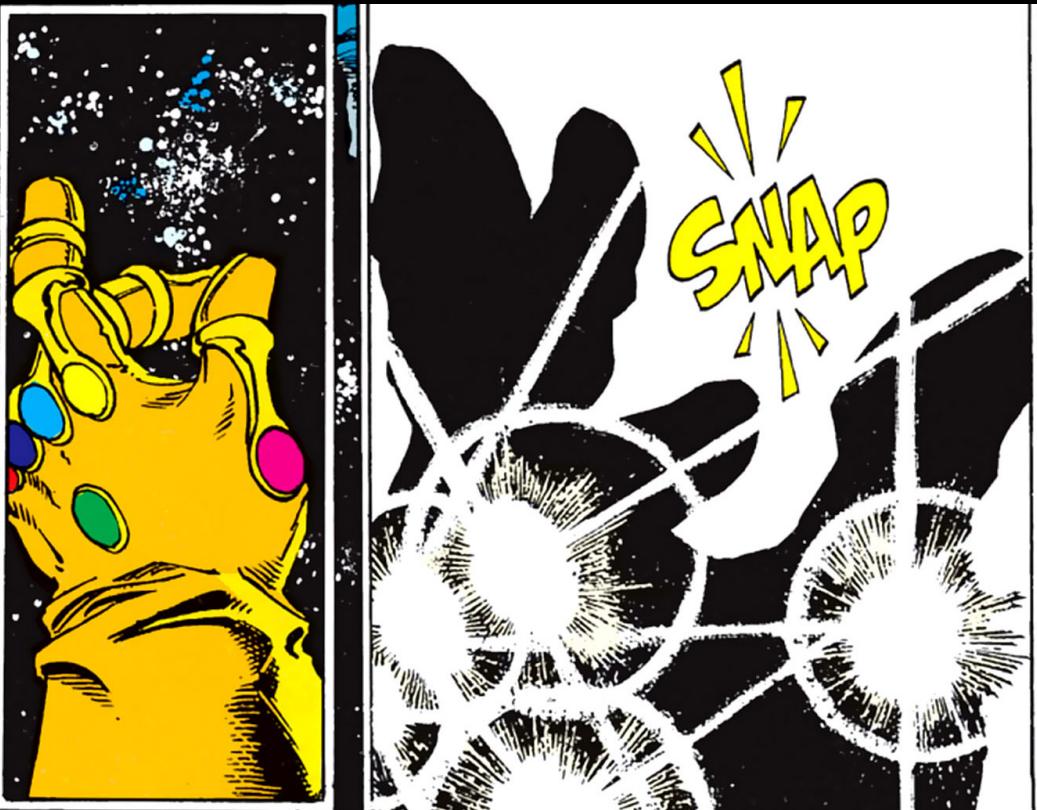
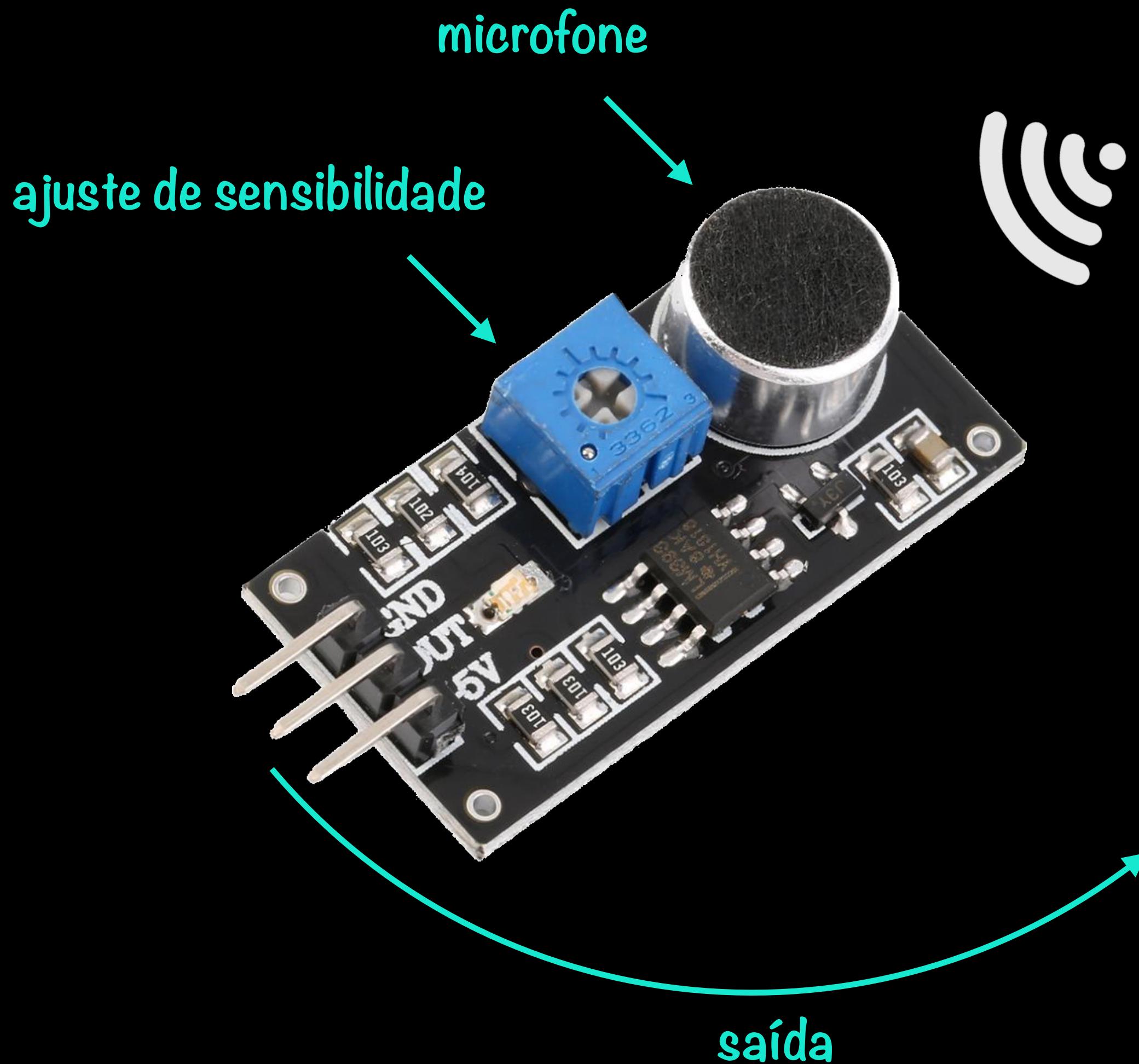
void setup () {
    pinMode(terra, OUTPUT);
    digitalWrite(terra, LOW);

    pinMode(campainha, OUTPUT);

    // tone não trava a execução do programa
    tone(campainha, 220.0, 500);
    // portanto, temos que esperar um pouco...
    delay(500);
    // ... antes de tocar a próxima frequência
    tone(campainha, 440.0, 500);
}
```



Sensor de Som



Funcionamento do Sensor de Som

```
int sensorDeSom = 19;  
void setup () {  
    Serial.begin(9600);  
    pinMode(sensorDeSom, INPUT);  
}  
  
void loop () {  
    if (digitalRead(sensorDeSom) == HIGH) {  
        Serial.println("som!");  
    }  
}
```

Essa abordagem tem vários
PROBLEMAS na prática...

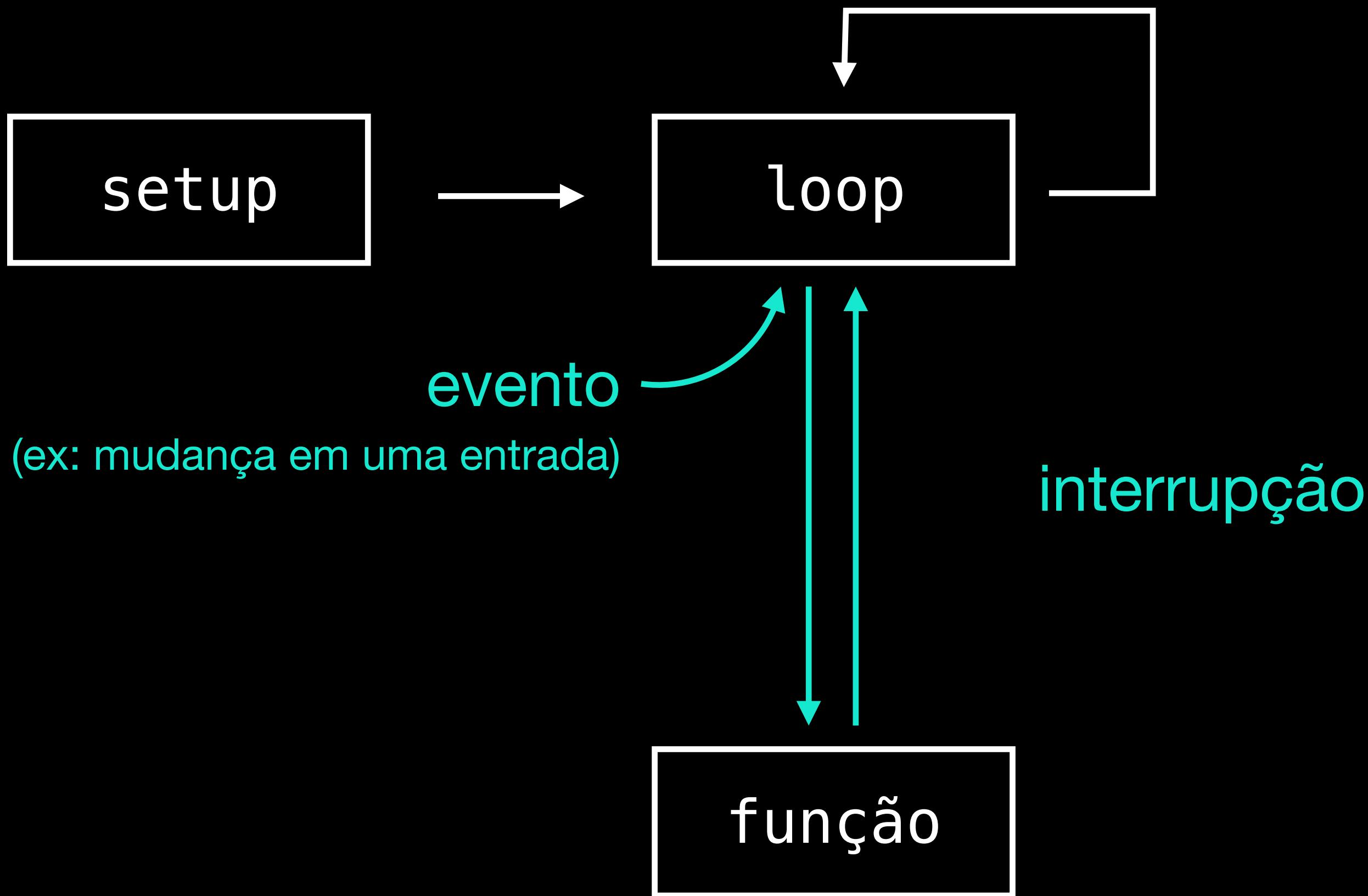


```
#include <ShiftDisplay.h>

ShiftDisplay display(4, 7, 8, COMMON_ANODE, 4, true);
int sensorDeSom = 19;
void setup () {
    Serial.begin(9600);
    pinMode(sensorDeSom, INPUT);
    display.set(1234);
}

void loop () {
    display.update(); ← demora 4 milissegundos
    if (digitalRead(sensorDeSom) == HIGH) {
        Serial.println("som!");
    }
}
```

← não dá tempo de detectar
o pulso de 1 milissegundo!



Interrupção do Loop

Modelo de Arduíno	Pinos
Uno, Nano, Mini	2, 3
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21
Micro, Leonardo	0, 1, 2, 3, 7
Zero	all digital pins, except 4
MKR1000 Rev.1	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Due, 101	todos

Pinos com Recurso de Interrupção

```
int origem = digitalPinToInterrupt(pino);  
attachInterrupt(origem, funcaoParaChamar, TIPO);
```

Tipo	Evento
RISING	entrada passa de LOW para HIGH
FALLING	entrada passa de HIGH para LOW
CHANGE	qualquer mudança na entrada

Função para Configurar Interrupção

```
int sensorDeSom = 19;  
void setup () {  
    Serial.begin(9600);  
    int origem = digitalPinToInterrupt(sensorDeSom);  
    attachInterrupt(origem, somDetectado, RISING);  
    ...  
}  
  
void loop () {  
    display.update();  
}  
  
void somDetectado () {  
    Serial.println("som!");  
}
```

Ok, mas ainda tem
um **probleminha...**





SNAP

A screenshot of a terminal window showing multiple 'som!' messages. The window title is '/dev/cu.usbmodem1421 (Arduino/Genuino Mega or Mega 2560)'. The text area contains:

```
som!
som!
som!
```

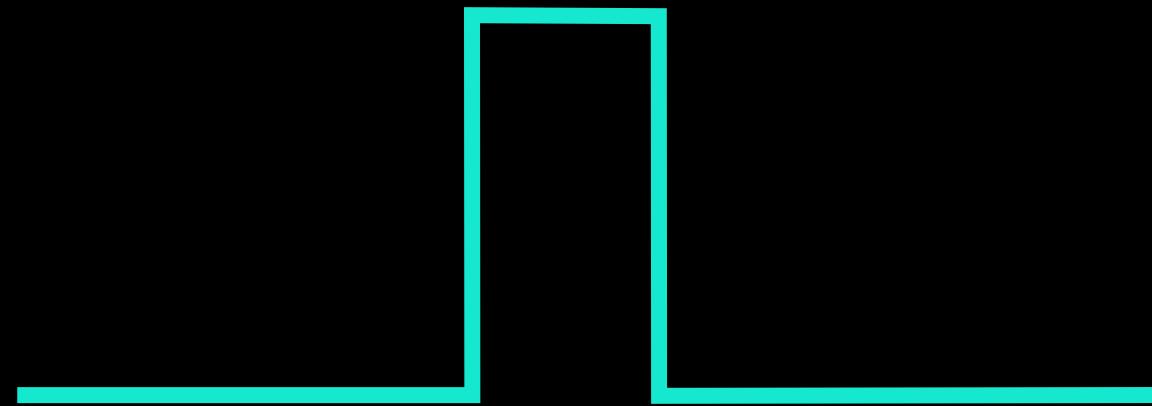
The bottom bar includes checkboxes for 'Auto-rolagem', 'Nenhum final-de-linha', '9600 velocidade', and 'Delete a saída'.

Por que um único estalo
às vezes gera várias interrupções?

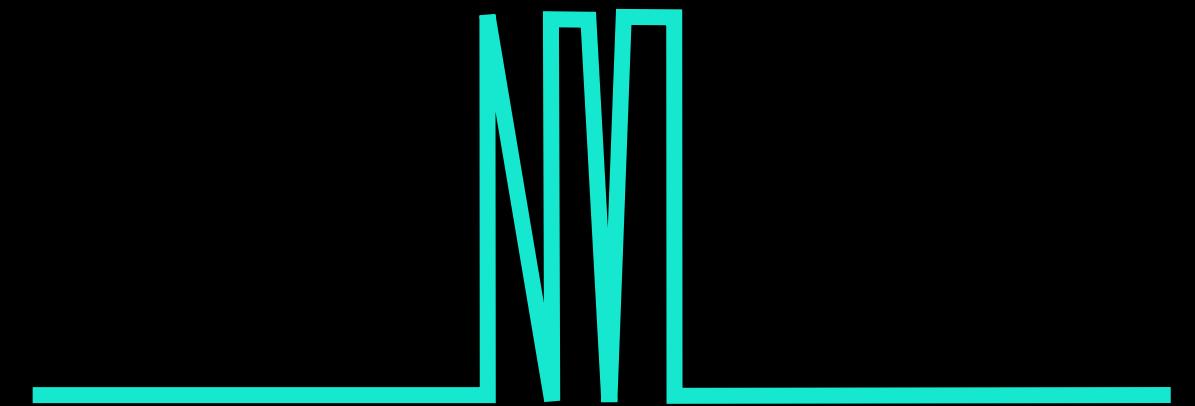


Problema 2: Detecção Duplicada

em teoria



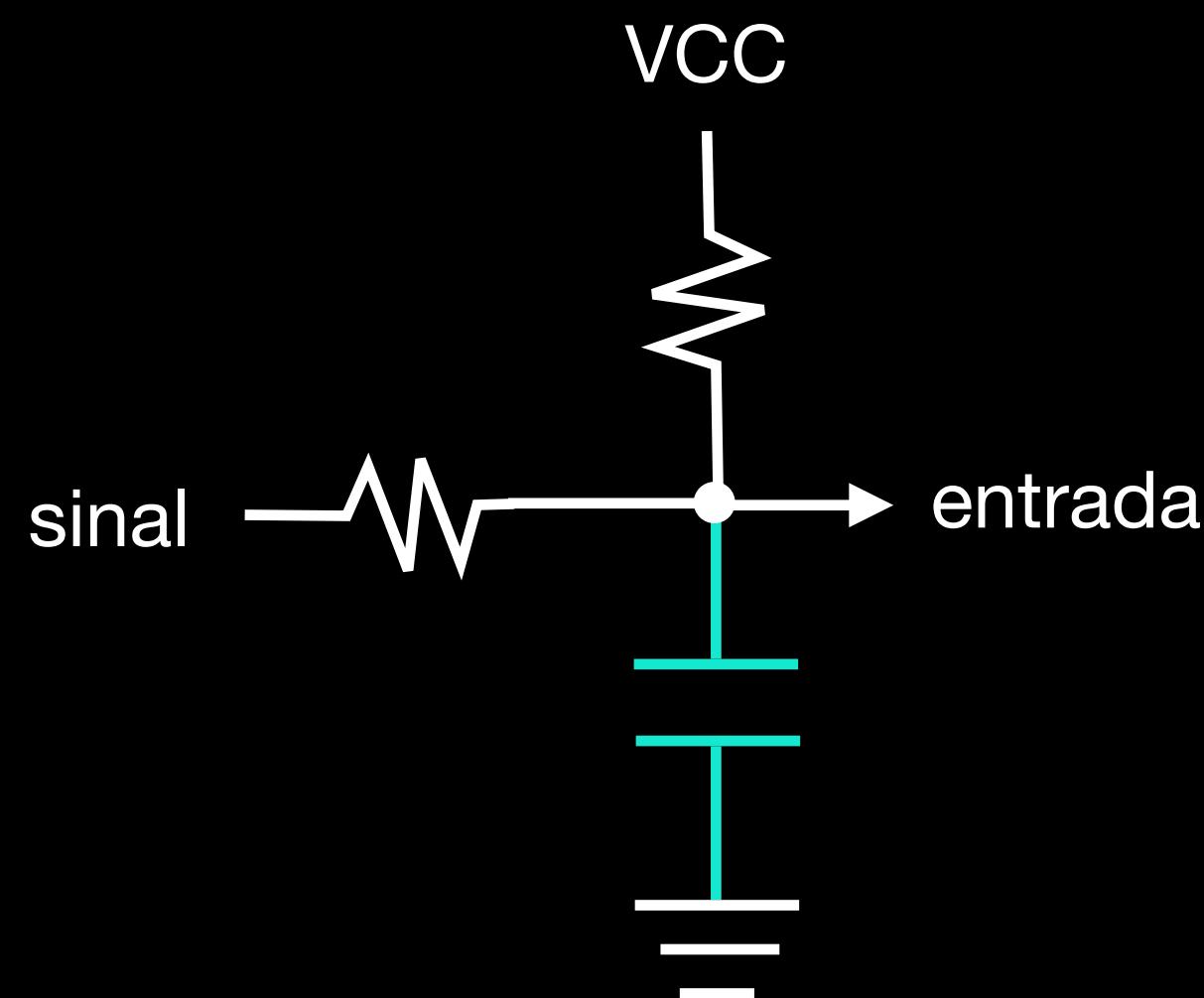
na prática



"bounce"

Entrada com "Bounce"

"debounce" via hardware



"debounce" via software

interrupção detectada

interrupção anterior
foi há **mais de uns 10 ms?**

sim

não

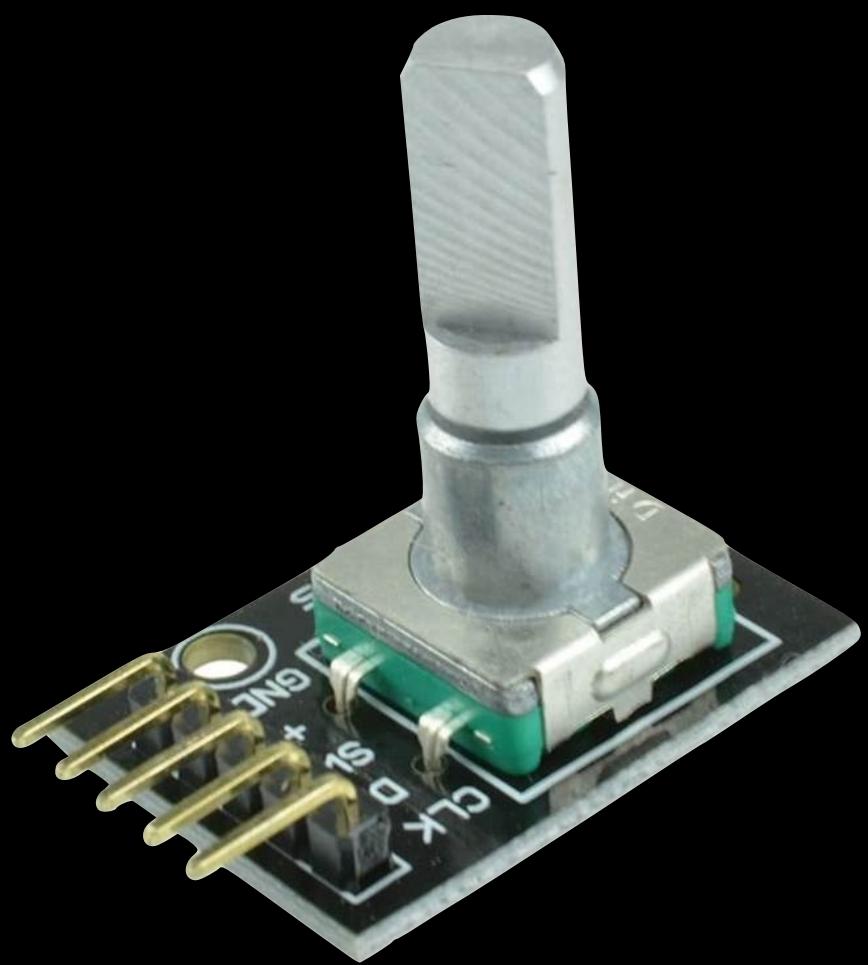
processa

ignora

Remoção do "Bounce"

```
unsigned long instanteAnterior = 0;  
...  
void setup () {  
    int origem = digitalPinToInterrupt(sensorDeSom);  
    attachInterrupt(origem, somDetectado, RISING);  
    ...  
}  
  
void loop () {  
    ...  
}  
  
void somDetectado () {  
    unsigned long instanteAtual = millis();  
    if (instanteAtual > instanteAnterior + 10) {  
        Serial.println("som!");  
        instanteAnterior = instanteAtual;  
    }  
}
```



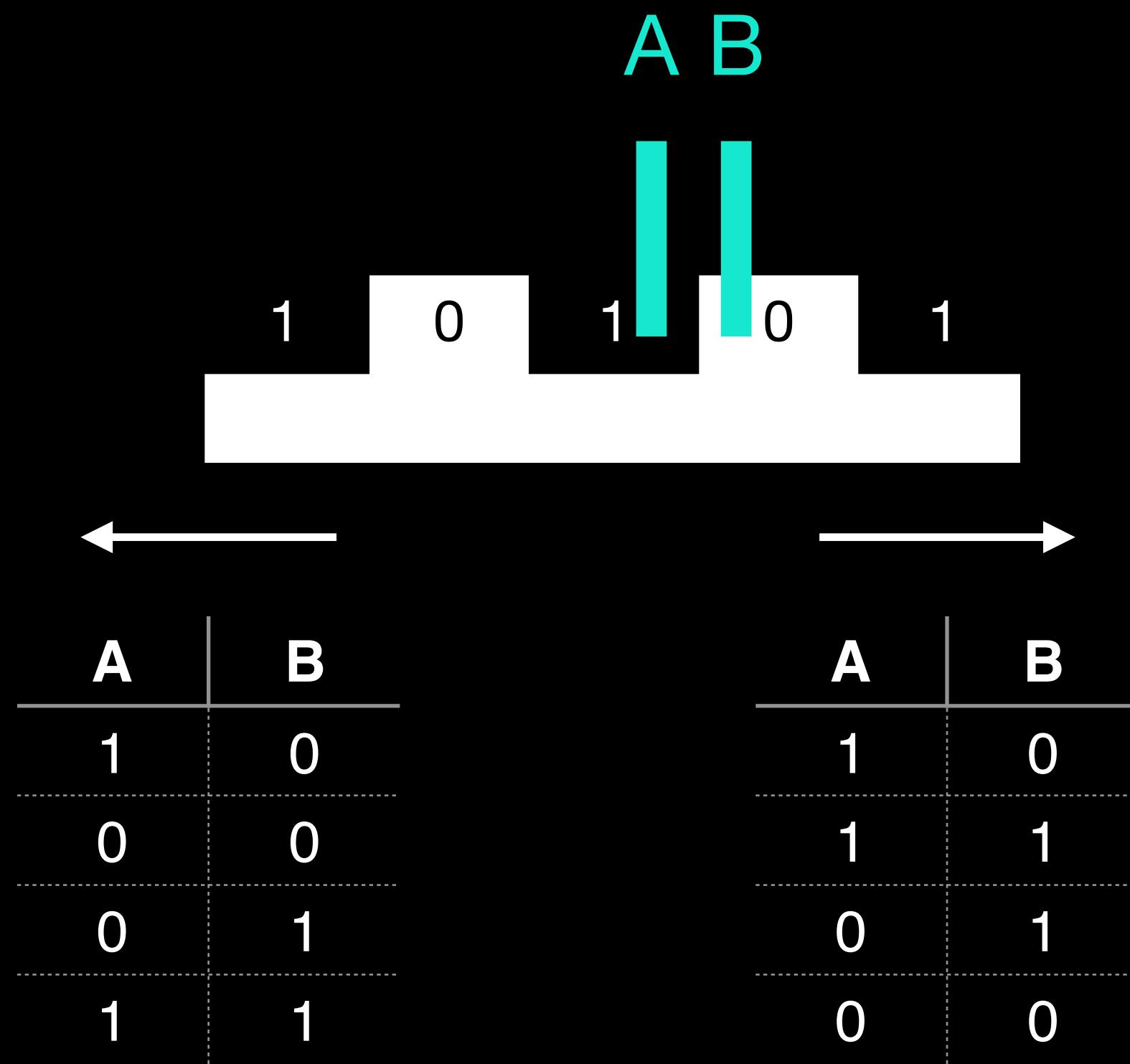
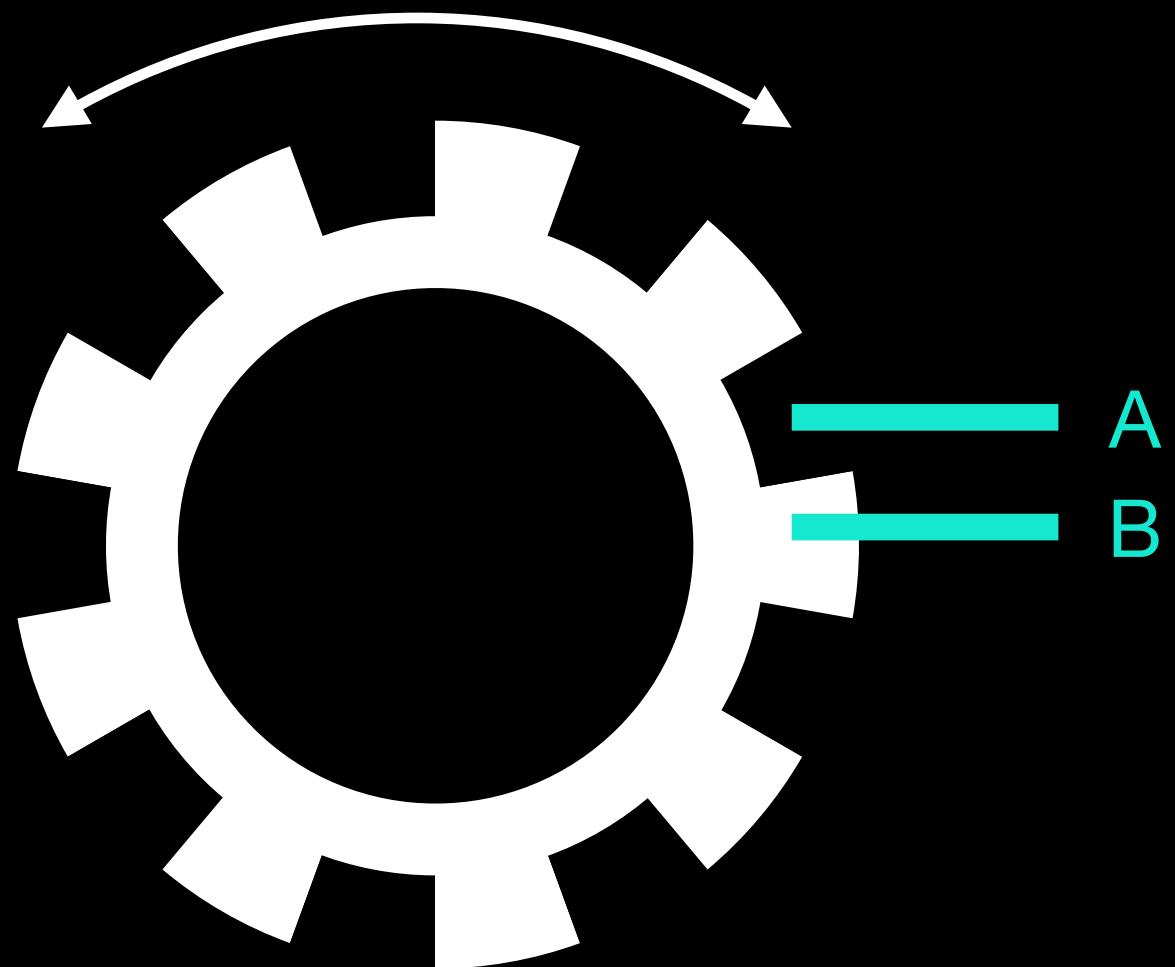


Encoder Rotativo

Gira Infinitamente



Exemplo de Encoder Rotativo para Controle Sonoro



Funcionamento Interno de um Encoder

The screenshot shows a web browser window with the URL www.mathertel.de/Arduino/RotaryEncoderLibrary.aspx in the address bar. The page itself has a header with the logo "mathertel.de" and navigation links for "Arduino", "Diff", "AJAXEngine", and "OpenAjax". Below the header is a breadcrumb trail: "www.mathertel.de > Arduino Projects > Arduino Rotary Encoder Library". The main content features a large, bold heading: "A Library for the Arduino environment for using a rotary encoder as an input." To the right of this heading is a vertical sidebar with links to "Sitemap", "Impressum", "License", and "Agreement". The main text area describes the purpose of the library and its development process, mentioning GitHub for sources.

Here you can find an Arduino compatible library for using rotary encoders.

I was searching a library for using a rotary encoder in my latest project and found a lot of information on this topic but none of the existing libraries did immediately match my expectations so I finally built my own.

This article likes to explain the software mechanisms used in detail so you can understand the coding and might be able to adjust it to your needs if you like. There are various aspects when writing a library for rotary encoders and you can also find a lot of the sources I analyzed at the bottom of this article.

Download

You can download the library and examples directly from the github repository that you can find at:

- <https://github.com/mathertel/RotaryEncoder>

Use the "Download zip file" button to get all the files and put them into your Sketches /libraries folder.

Rotary Encoder signals

The signals a rotary encoder produces (and what can be handled by this library) are based on a 2-bit gray code available on 2 digital data signal lines. The first line goes from 0 to 1 via 0 for the signal and

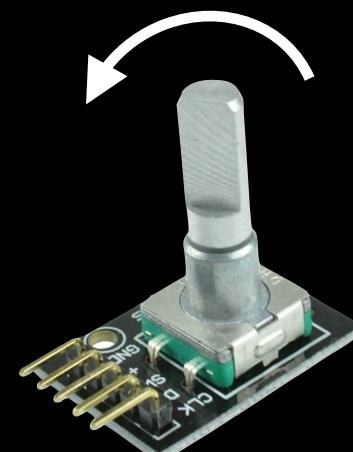
verifica estado das saídas, para contar os giros
encoder.tick()

retorna a posição atual de giro

int x = encoder.getPosition()



0 → 1 → 2 → 3 → 4 → ...



3 → 2 → 1 → 0 → -1 → ...

```

#include <RotaryEncoder.h>

RotaryEncoder encoder(20, 21);
int posicaoAnterior = 0;
void setup() {
    Serial.begin(9600);
}

void loop() {
    encoder.tick();
    int posicao = encoder.getPosition();
    if (posicao != posicaoAnterior) {
        Serial.println(posicao);
        posicaoAnterior = posicao;
    }
}

```



Sinto que isso vai dar
PROBLEMA de novo...



Exemplo da RotaryEncoder

```

#include <RotaryEncoder.h>
#include <ShiftDisplay.h>

RotaryEncoder encoder(20, 21);
ShiftDisplay display(4, 7, 8, COMMON_ANODE, 4, true);
int posicaoAnterior = 0;
void setup() {
    Serial.begin(9600);
    display.set(1234);
}

void loop() {
    display.update();
    encoder.tick();           ← demora 4 milissegundos
    int posicao = encoder.getPosition();
    if (posicao != posicaoAnterior) {
        Serial.println(posicao);
        posicaoAnterior = posicao;
    }
}

```



```

#include <RotaryEncoder.h>
#include <ShiftDisplay.h>

RotaryEncoder encoder(20, 21);
ShiftDisplay display(4, 7, 8, COMMON_ANODE, 4, true);
int posicaoAnterior = 0;
void setup() {
    int origem1 = digitalPinToInterrupt(20);
    attachInterrupt(origem1, tickDoEncoder, CHANGE);
    int origem2 = digitalPinToInterrupt(21);
    attachInterrupt(origem2, tickDoEncoder, CHANGE);
}

void tickDoEncoder() {
    encoder.tick();
}

void loop() {
    display.update();
    int posicao = encoder.getPosition();
    if (posicao != posicaoAnterior) {
        ...
    }
}

```



Exemplo da RotaryEncoder com Interrupção

Resumo da Ópera

Funcionalidade

Campainha Passiva
[documentação](#)

Interrupção
[documentação](#)

Contagem
de Tempo
[documentação](#)

Encoder Rotativo
[documentação](#)

Comandos

```
int campainhaPassiva = 5;  
pinMode(campainhaPassiva, OUTPUT);  
int frequencia = 220; int duracaoEmMs = 500;  
tone(campainhaPassiva, frequencia);  
tone(campainhaPassiva, frequencia, duracaoEmMs);  
noTone(campainhaPassiva);
```

```
int sensorDeSom = 19;  
pinMode(sensorDeSom, INPUT);  
int origem = digitalPinToInterrupt(sensorDeSom);  
attachInterrupt(origem, minhaFuncao, RISING);  
unsigned long instanteAnteriorDeDeteccao = 0;
```

```
if (millis() > instanteAnteriorDeDeteccao + 10) {  
    instanteAnteriorDeDeteccao = millis();  
}
```

```
#include <RotaryEncoder.h>  
RotaryEncoder encoder(20, 21);  
int origem1 = digitalPinToInterrupt(20);  
attachInterrupt(origem1, tickDoEncoder, CHANGE);  
int origem2 = digitalPinToInterrupt(21);  
attachInterrupt(origem2, tickDoEncoder, CHANGE);  
encoder.tick(); int posicao = encoder.getPosition();  
encoder.setPosition(posicao);
```

Funcionalidade

Revisão de C++

Comandos

```
int inteiro = 2; float decimal = 4.5; bool booleano = true;  
char texto[] = "Olá"; int listaDeInteiros[] = {1, 2, 3, 4};  
  
if (x > 0 && y > 0) {  
    z = 1;  
}  
else if (x < 0 || y < 0) {  
    z = 2;  
}
```

```
for (int i = 0; i < 5; i++) {  
    Serial.println(i);  
}  
float soma (float x) {  
    return x + 2;  
}
```

Print Serial

```
Serial.begin(9600); Serial.println("Olá"); Serial.println(2);
```

Escrita/Leitura documentação

```
int led = 13; pinMode(led, OUTPUT); digitalWrite(led, LOW);  
int campainha = 3; digitalWrite(campainha, HIGH);  
int botao = A1; pinMode(botao, INPUT); digitalWrite(botao) == LOW
```

GButton documentação

```
#include <GButton.h>  
GButton botao(A1); botao.isPressed(); botao.process();  
botao.setPressHandler(funcao); botao.setReleaseHandler(funcao);
```

ShiftDisplay documentação

```
#include <ShiftDisplay.h>  
ShiftDisplay display(4, 7, 8, COMMON_ANODE, 4, true);  
ShiftDisplay display(4, 7, 8, COMMON_CATHODE, 4, true);  
display.set(1234); display.set(4.21, 2); display.set("Erro");  
display.update(); display.show(1000); display.changeDot(0, true)
```

Timer1 documentação

```
#include <TimerOne.h>  
Timer1.initialize(1000000); Timer1.attachInterrupt(funcao);
```