



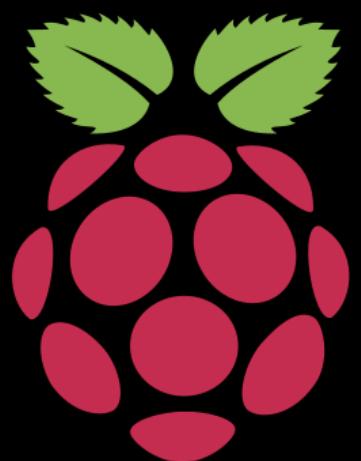
# Projeto 07

## 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

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

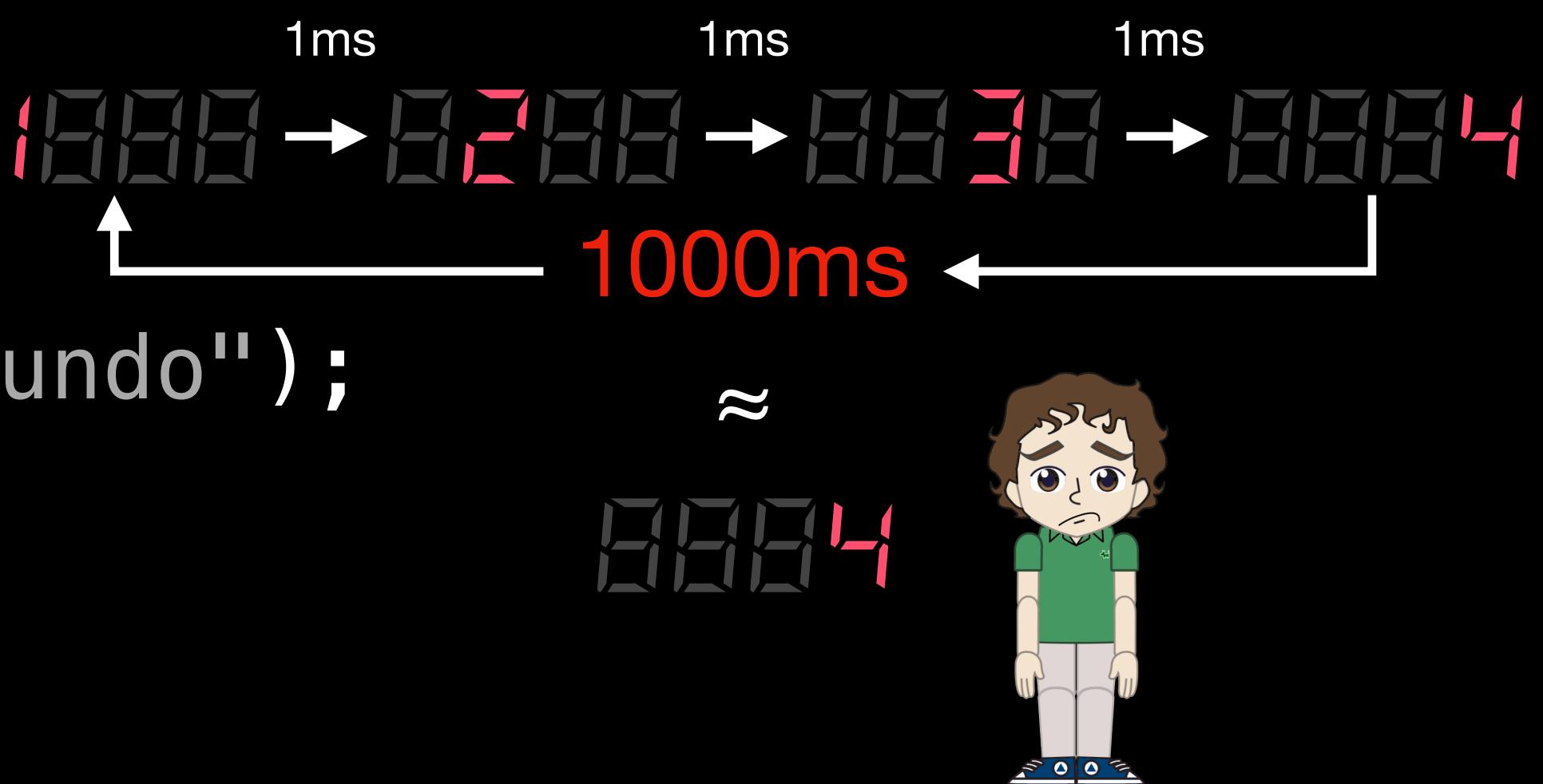


Usamos unsigned long para poder contar tempo sem esbarrar no limite de 2 bytes.

```
#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 int instanteAnterior = 0;

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

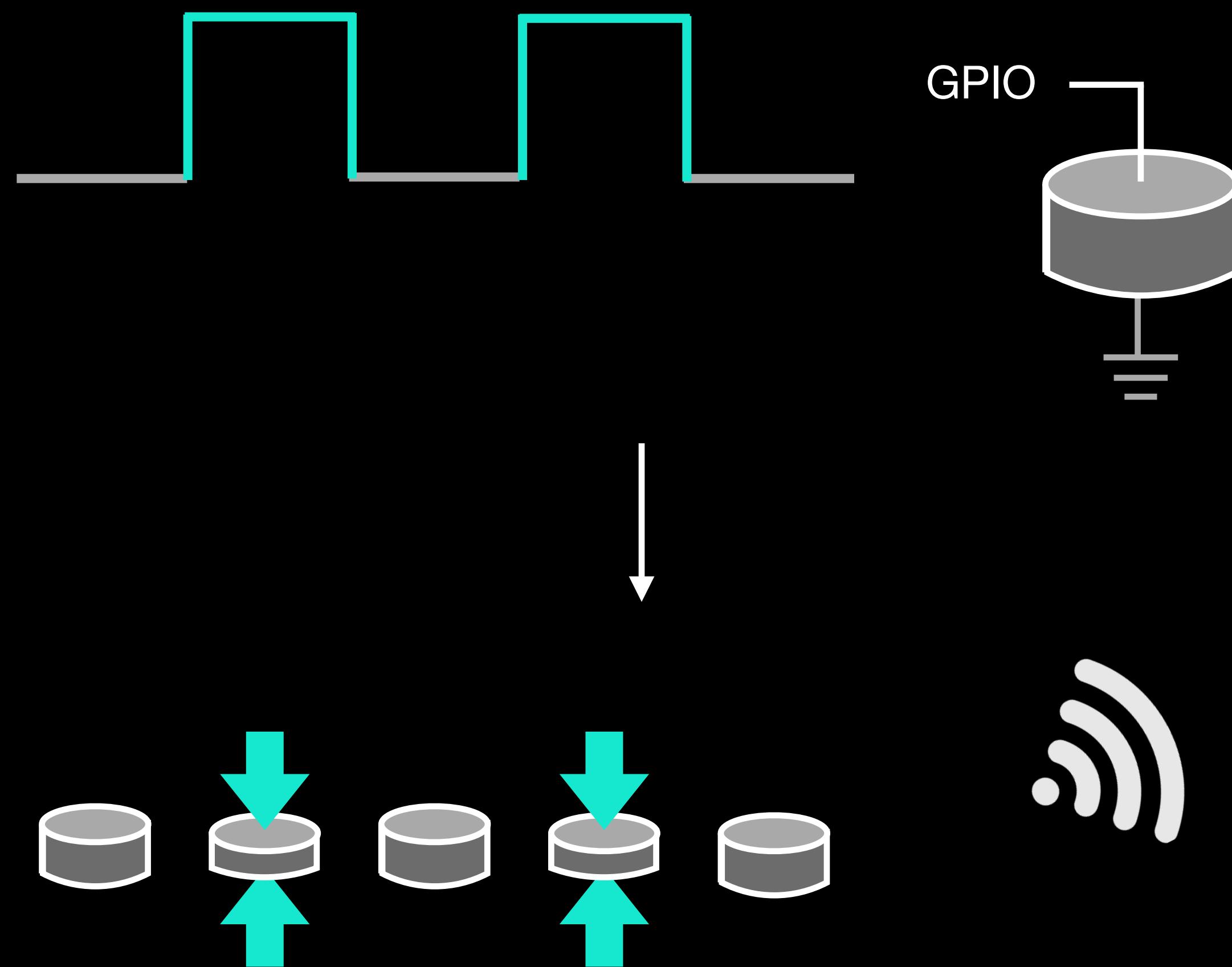
void loop () {
    display.update();

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

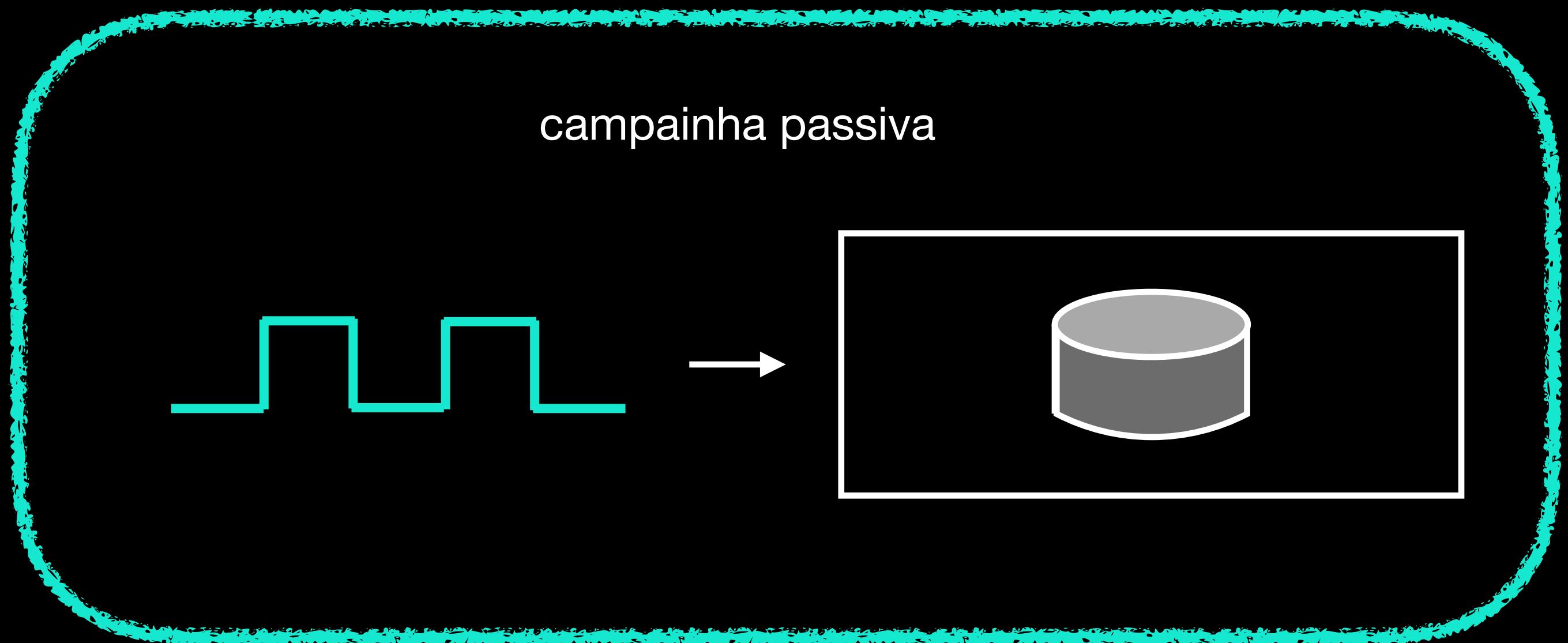
# Hardware



Campainha (Buzzer)



Geração de Som por Pulso

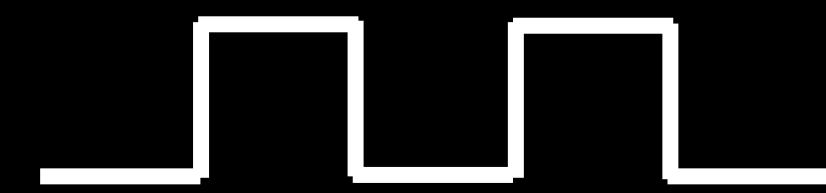
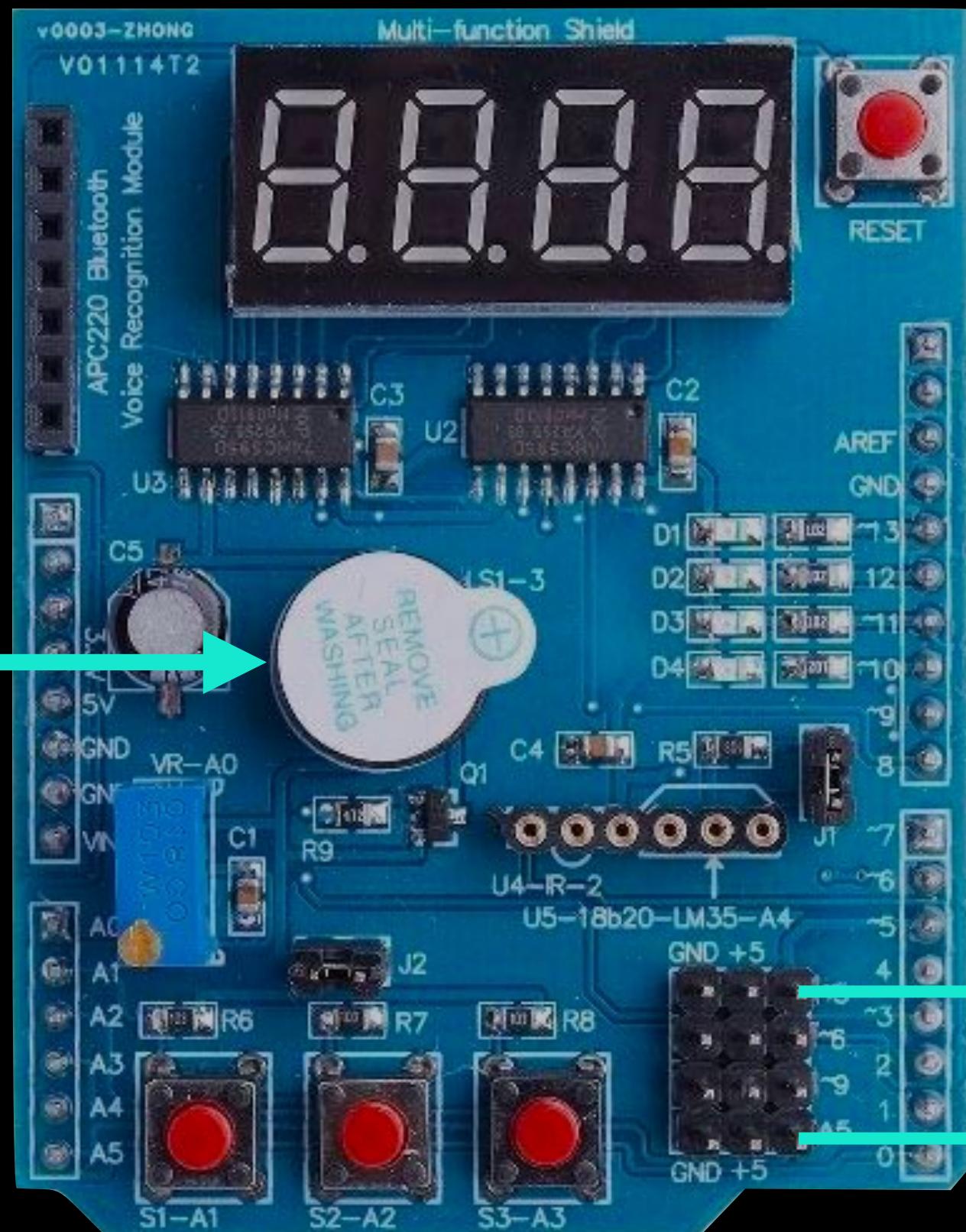


campainha ativa



Campainhas Ativas e Passivas

campainha  
ativa



campainha  
passiva

5

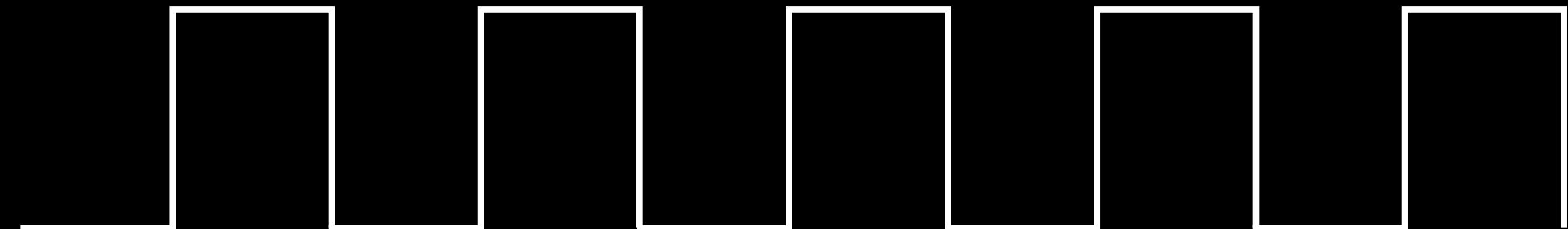
A5

LOW

Campainha Passiva Conectada a Duas Portas de Uso Geral

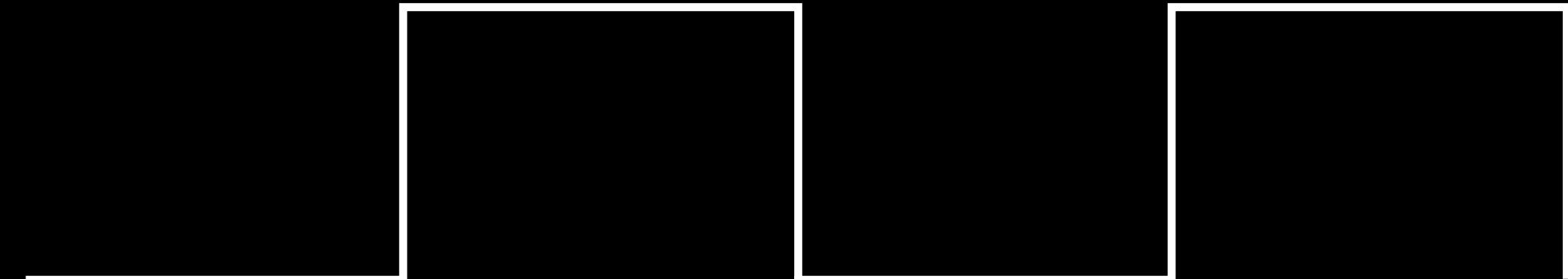
*pino* →  
*frequência em Hz* →  
tone(5, 440);

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);  
}
```

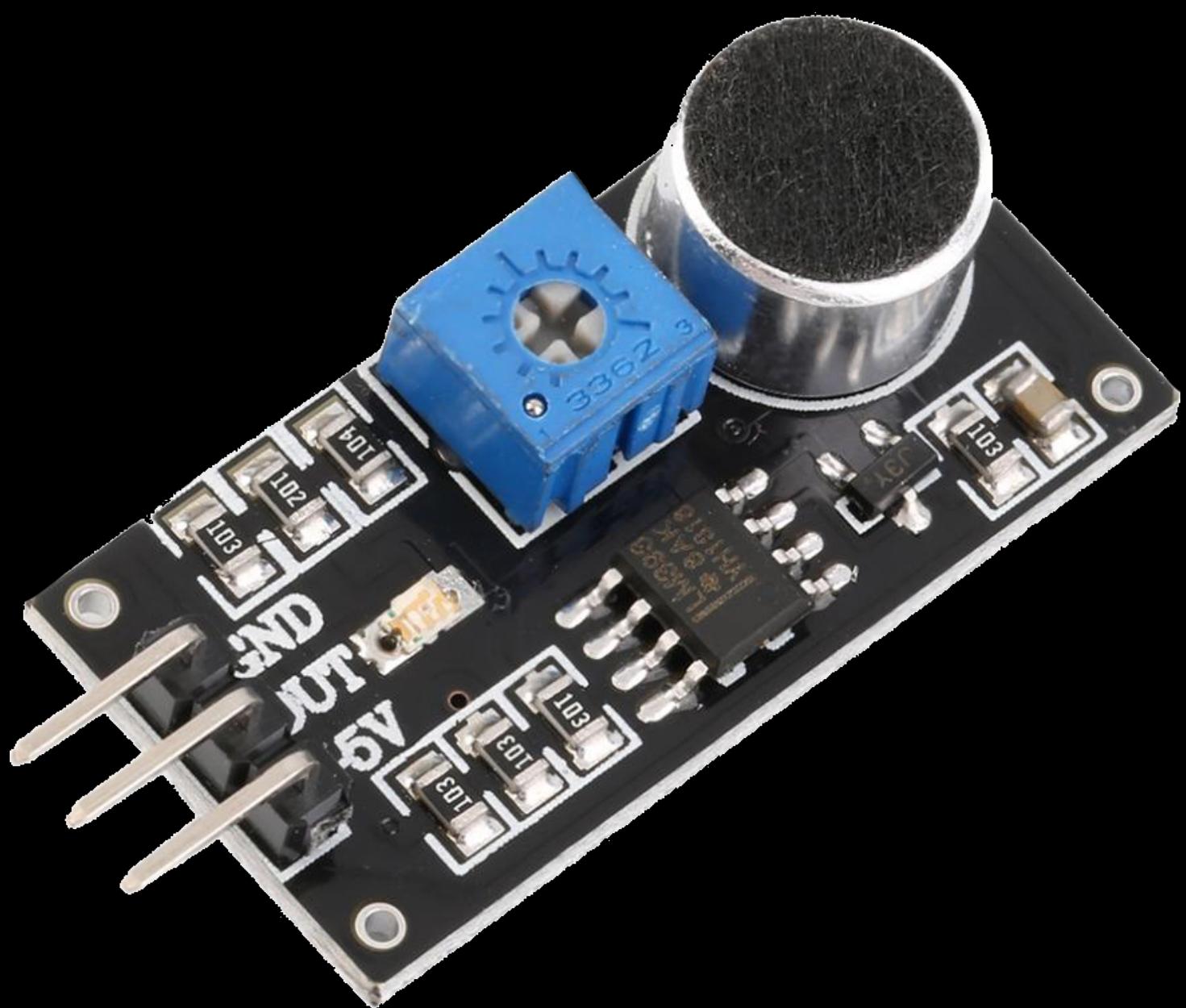
A função tone **não trava**  
a execução do programa



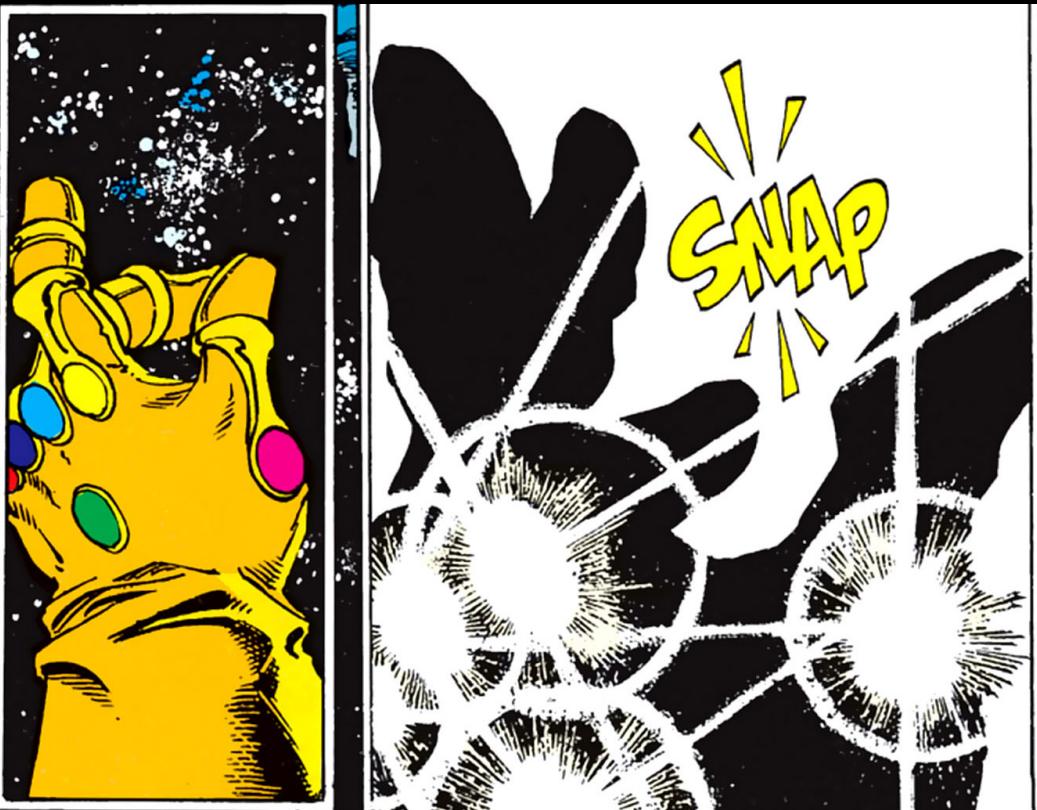
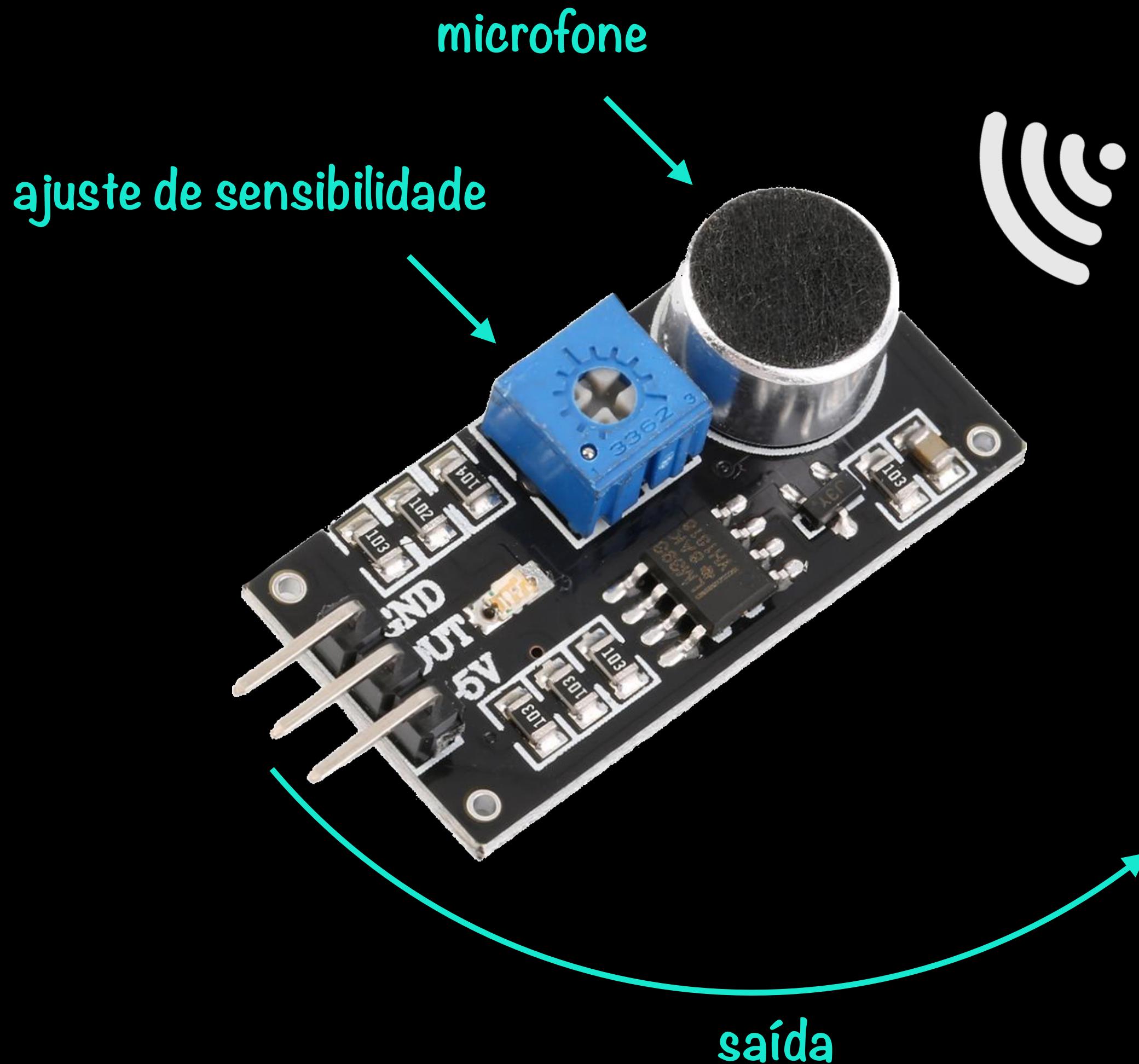
```
int terra = A5;
int campainha = 5;
void setup () {
    pinMode(terra, OUTPUT);
    digitalWrite(terra, LOW);

    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

Essa abordagem tem vários  
**problemas** na prática...

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

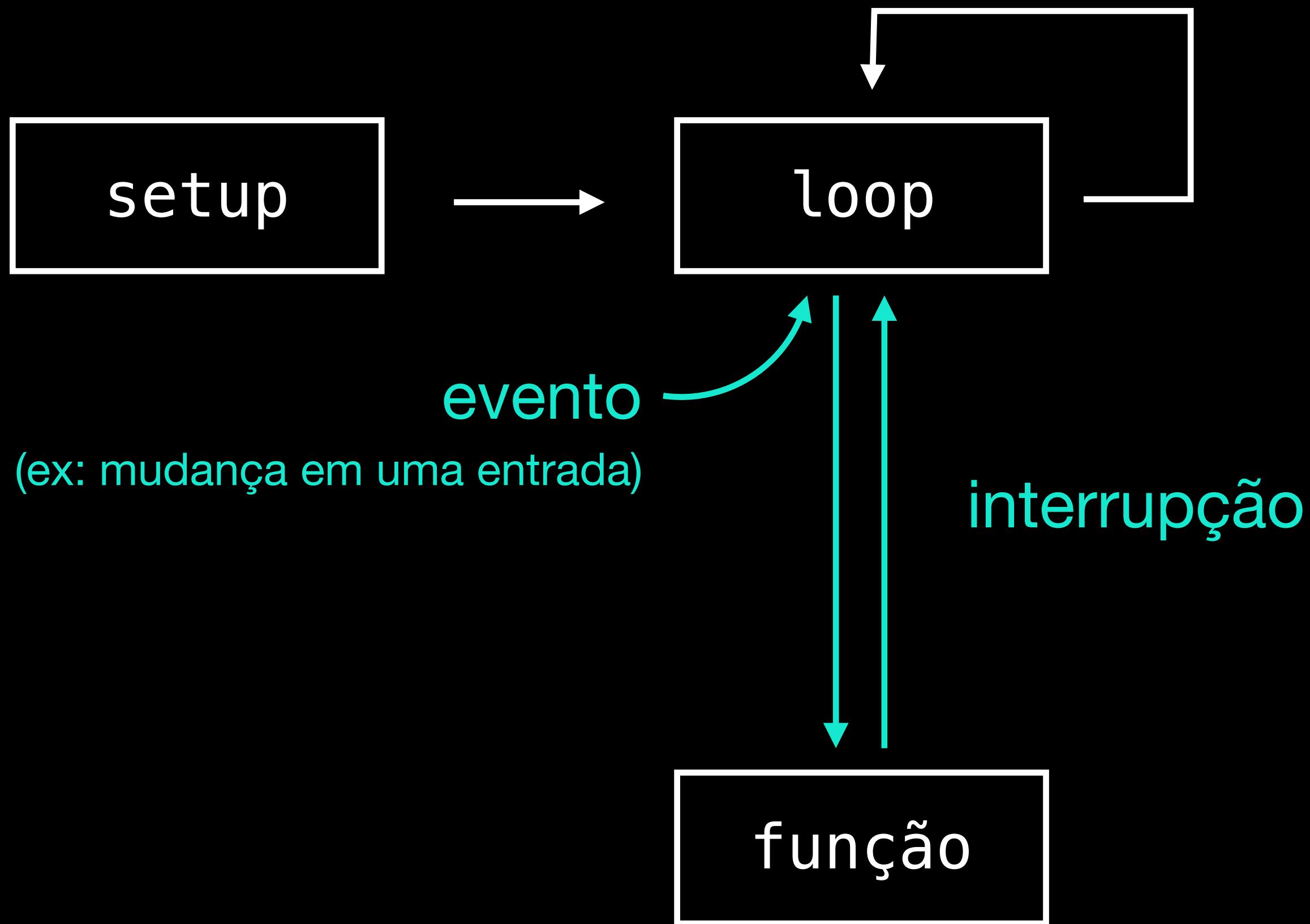


```
#include <ShiftDisplay.h>

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

<b>Modelo de Arduíno</b>	<b>Pinos</b>
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);
```

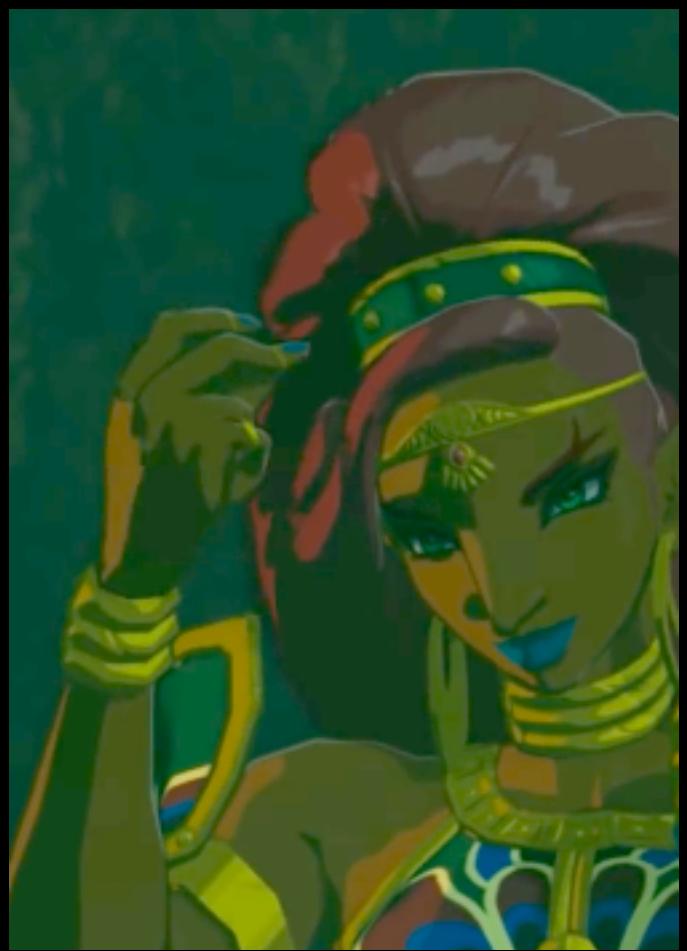
Tipo	Evento
RISING	entrada passa de LOW para HIGH
FALLING	entrada passa de HIGH para LOW
CHANGE	qualquer mudança na entrada

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...**





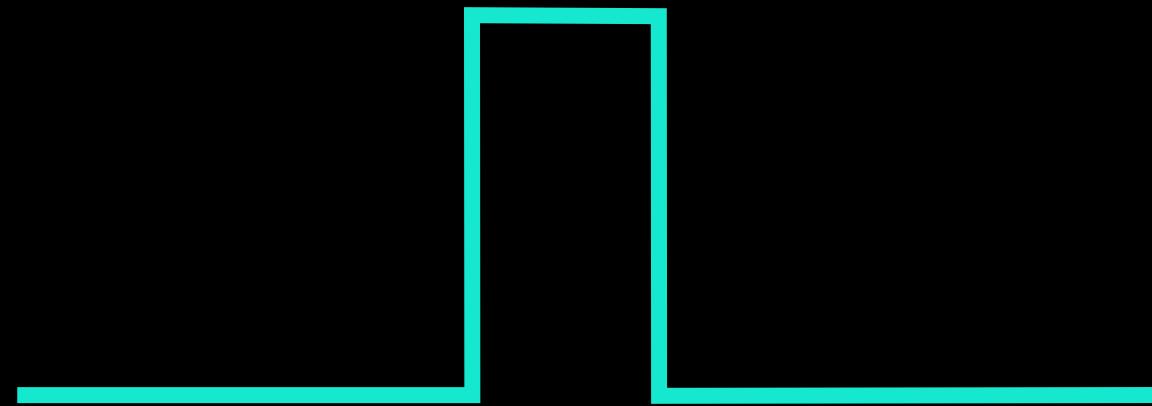
A screenshot of a terminal window titled '/dev/cu.usbmodem1421 (Arduino/Genuino Mega or Mega 2560)'. The window shows three identical messages: 'som!', 'som!', and 'som!'. At the bottom, there are several configuration options: 'Auto-rolagem' (checked), 'Nenhum final-de-linha', '9600 velocidade', and 'Delete a saída'.

Por que um único estalo  
às vezes gera várias interrupções?

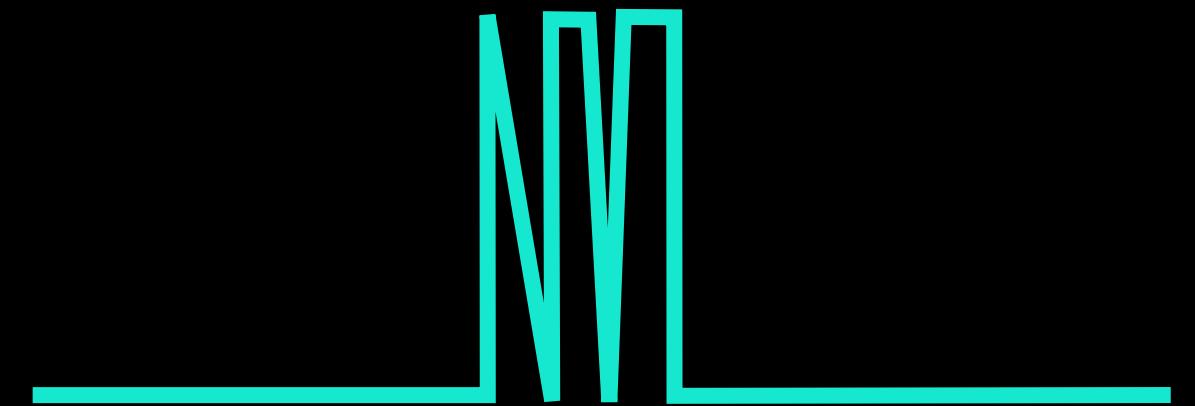


Problema 2: Detecção Duplicada

em teoria

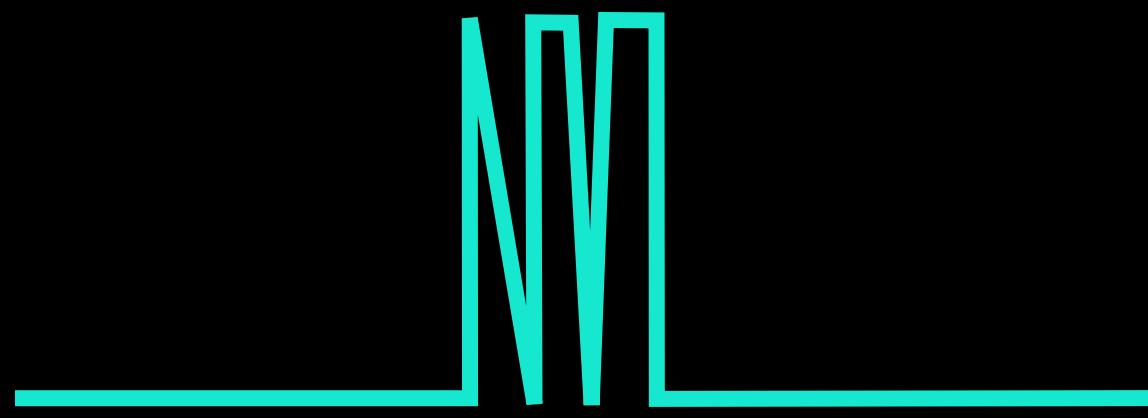


na prática

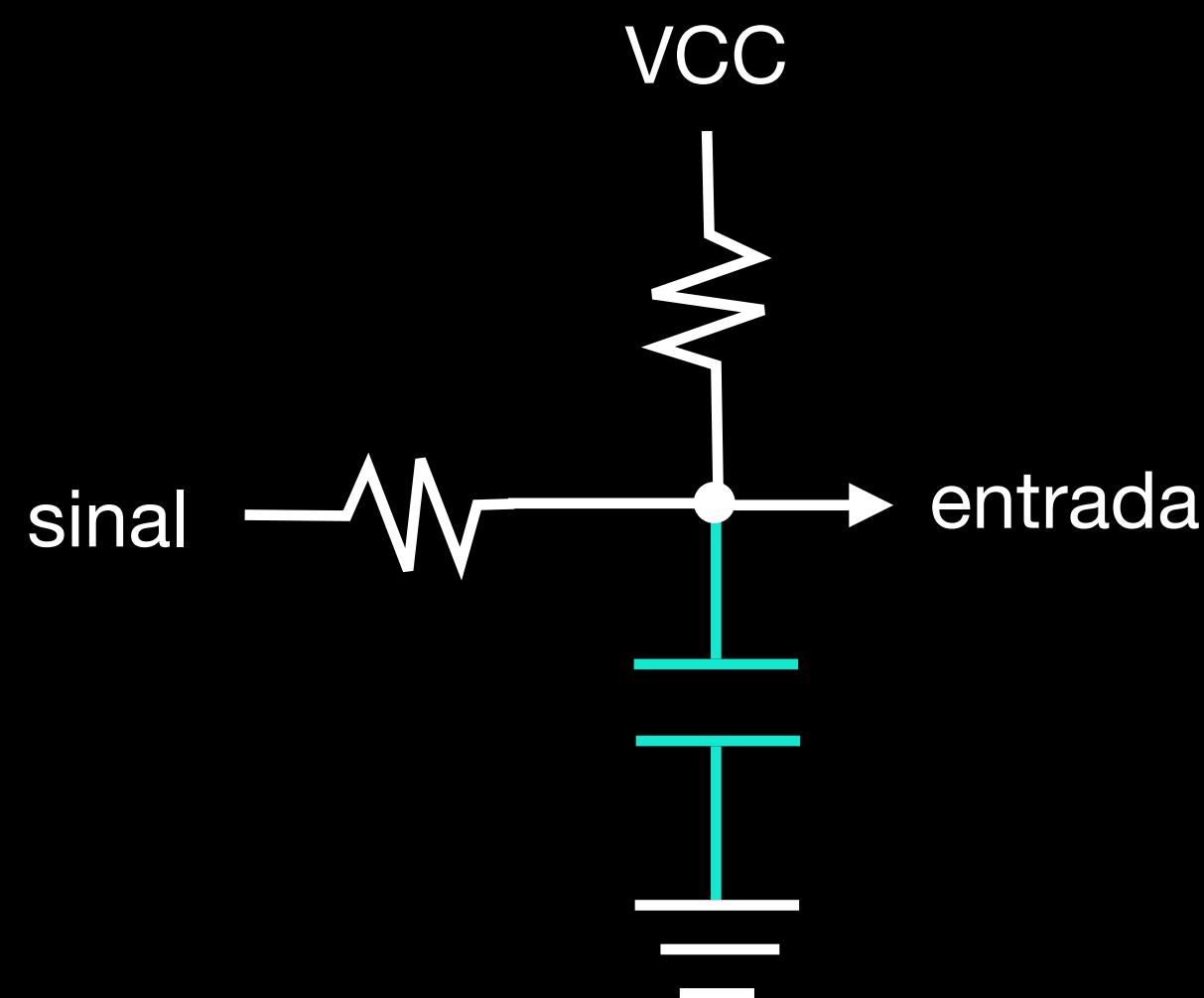


"bounce"

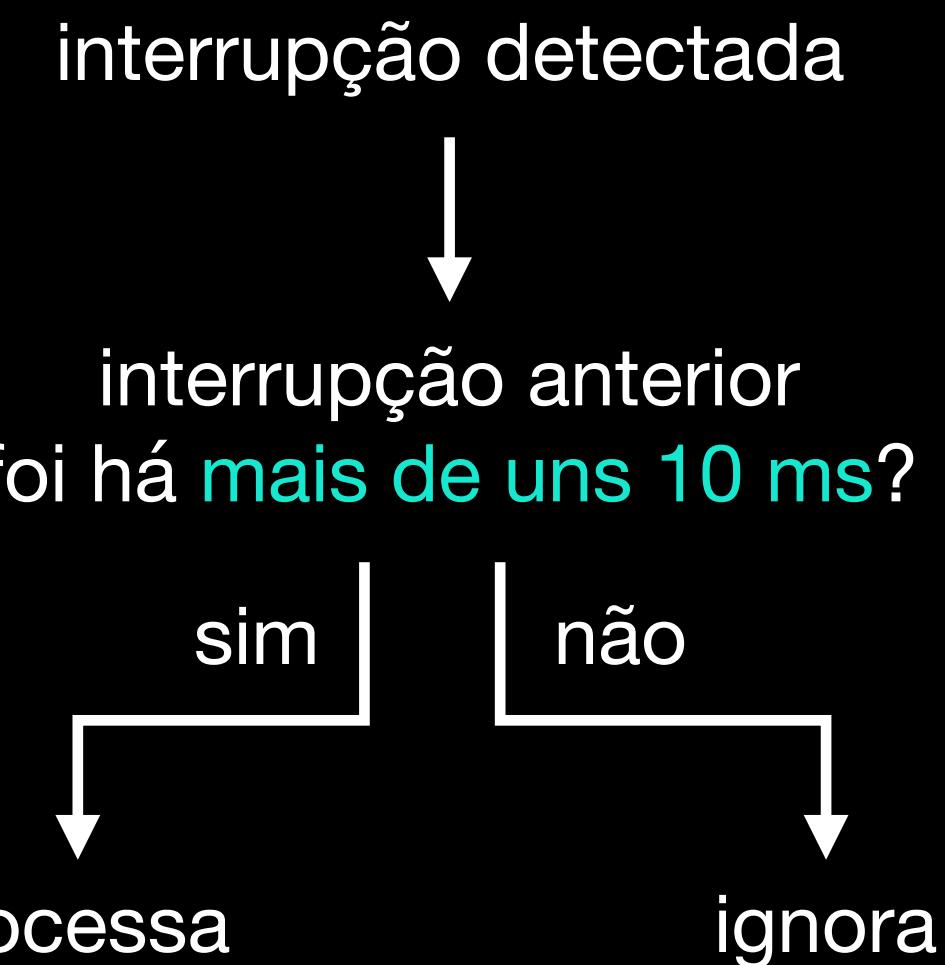
Entrada com "Bounce"



"debounce" via hardware



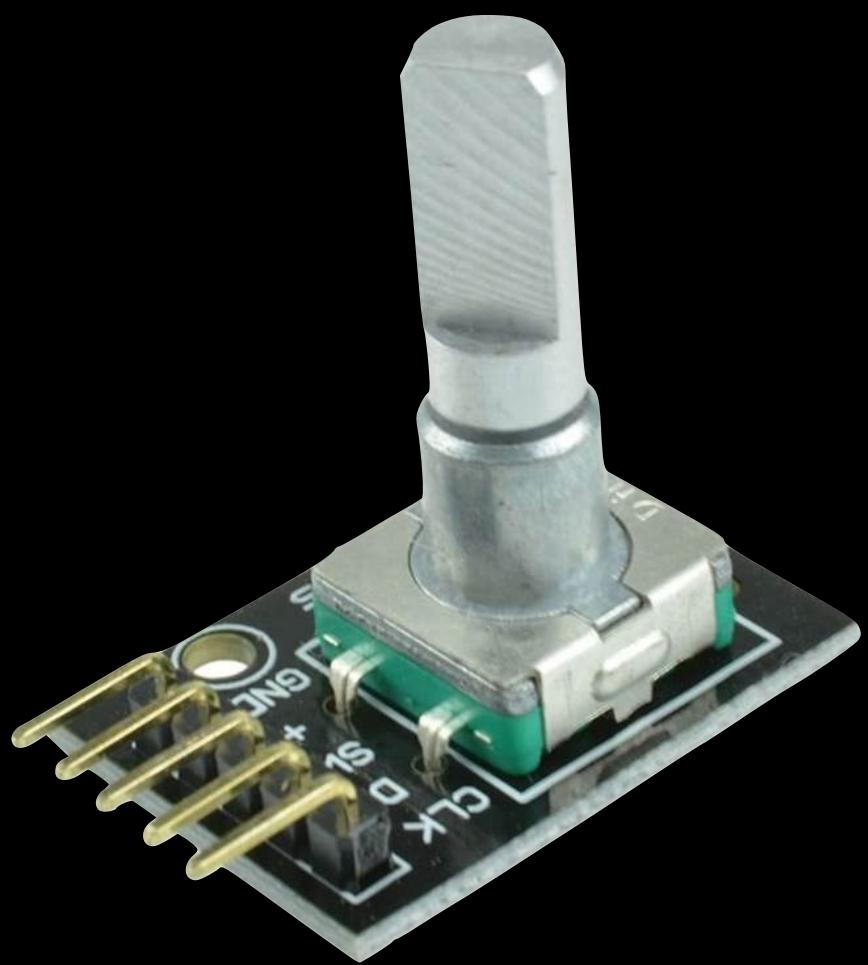
"debounce" via software



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;  
    }  
}
```



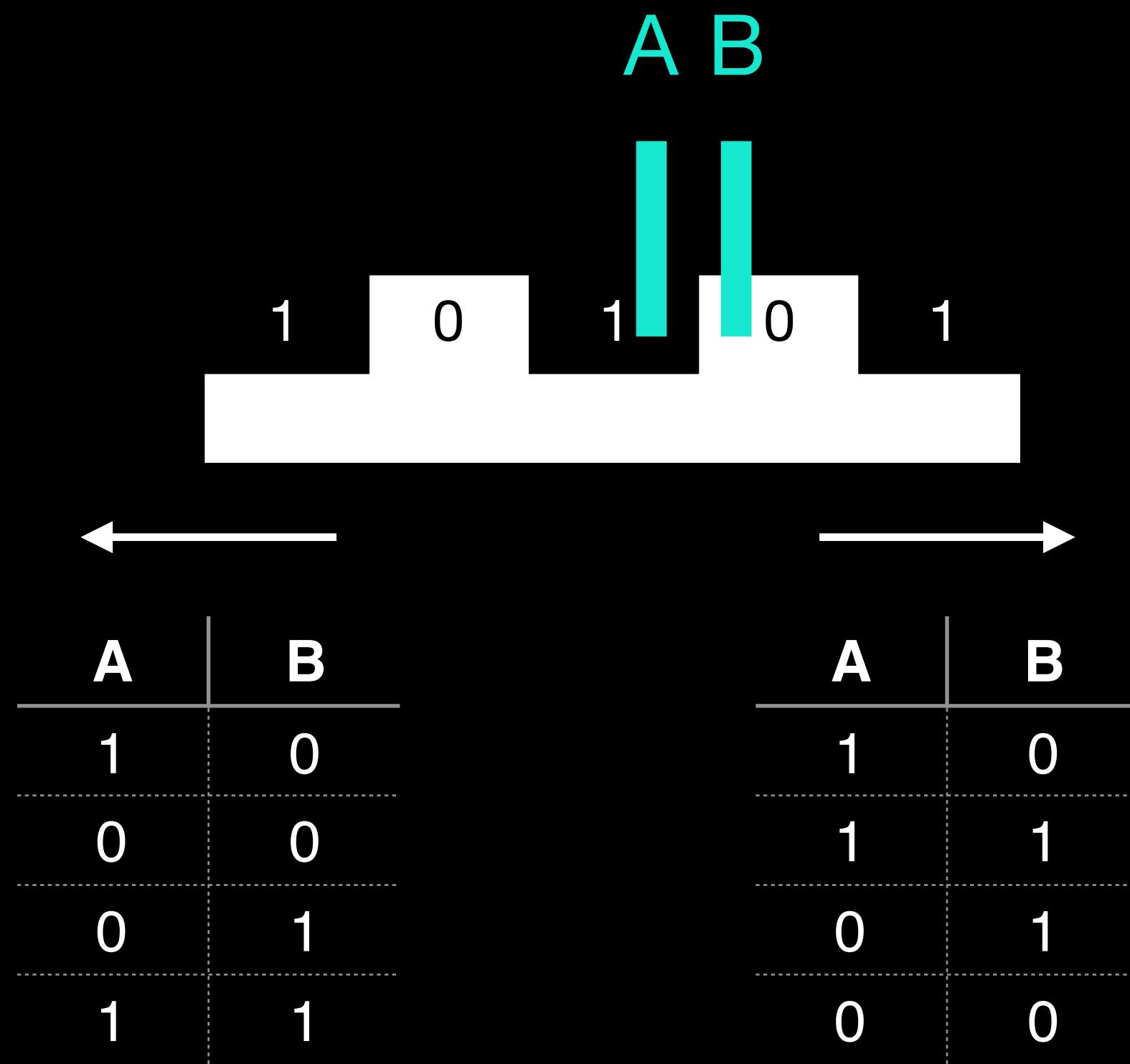
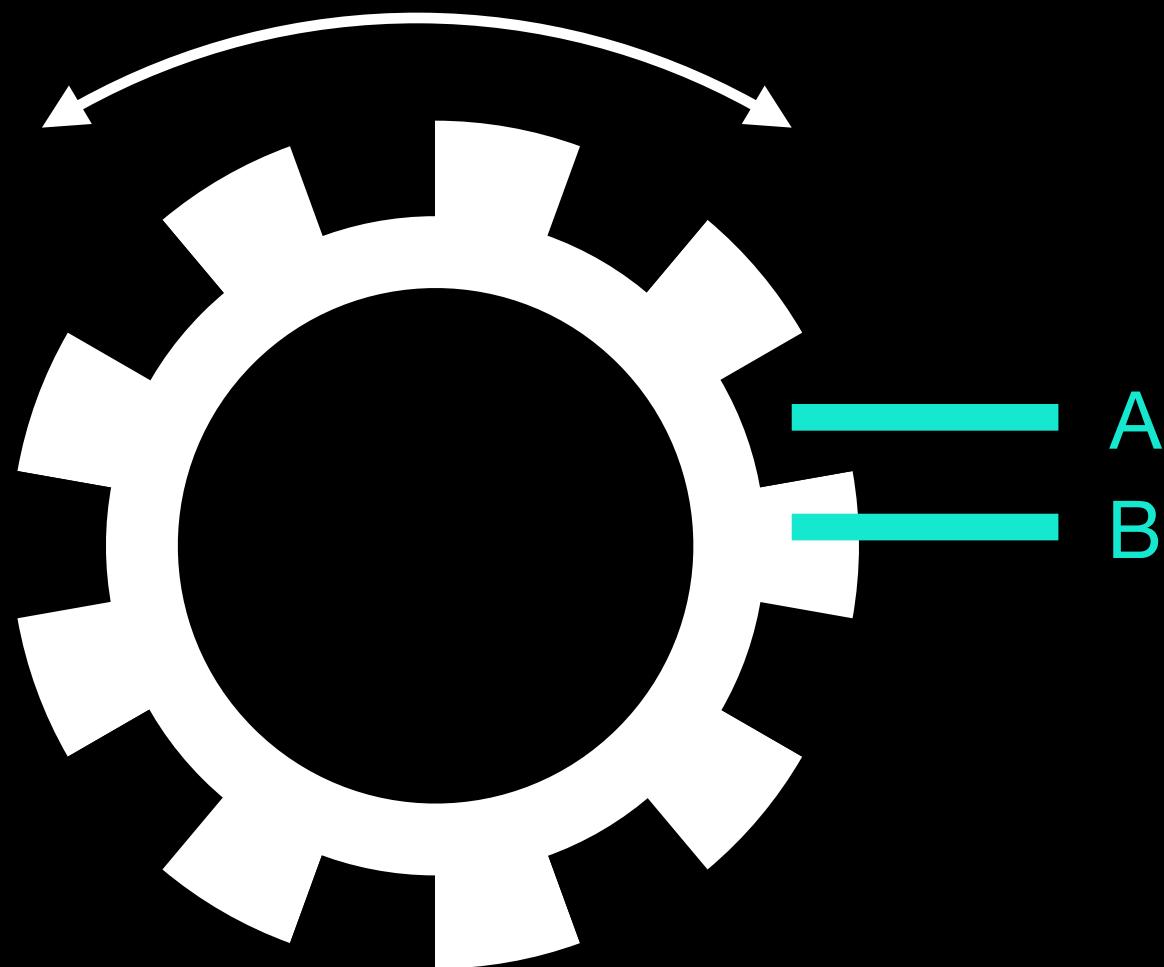


Encoder Rotativo

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](http://www.mathertel.de/Arduino/RotaryEncoderLibrary.aspx) in the address bar. The page is titled "A Library for the Arduino environment for using a rotary encoder as an input." It features a navigation menu with tabs for "Arduino" (selected), "Diff", "AJAXEngine", and "OpenAjax". Below the menu, a breadcrumb trail shows the current location: "www.mathertel.de > Arduino Projects > Arduino Rotary Encoder Library". On the right side of the page, there is a sidebar with links to "Sitemap", "Impressum", "License", and "Agreement". The main content area contains text explaining the library's purpose and how it was built, along with a "Download" section and a "Rotary Encoder signals" section.

**mathertel.de**

Arduino Diff AJAXEngine OpenAjax

www.mathertel.de > Arduino Projects > Arduino Rotary Encoder Library

# A Library for the Arduino environment for using a rotary encoder as an input.

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 two lines are denoted as Q<sub>0</sub> starting at 0 for the signal and

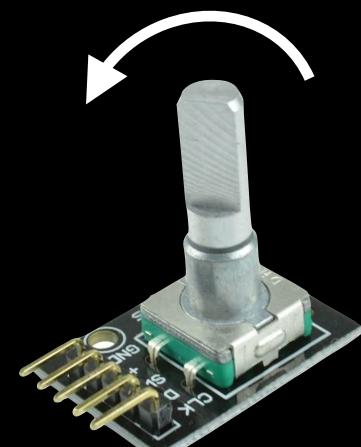
**verifica estado das saídas, para contar os giros**  
**encoder.tick()**

**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;
    }
}

```



Interferência da ShiftDisplay na 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() {
    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  
[acessar documentação](#)

## Comandos

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

Interrupção  
[acessar documentação](#)

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

Contagem  
de Tempo  
[acessar documentação](#)

```
unsigned long instanteAnteriorDeDeteccao = 0;  
  
if (millis() > instanteAnteriorDeDeteccao + 10) {  
    instanteAnteriorDeDeteccao = millis();  
}
```

Encoder Rotativo  
[acessar documentação](#)

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

## Funcionalidade

### Revisão de C++

### Print Serial

### Escrita/Leitura acessar documentação

### GButton acessar documentação

### ShiftDisplay acessar documentação

### Timer1 acessar documentação

## 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;  
}
```

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

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

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

```
#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);
```

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