

# Projeto 01

## Controle de Mídia

Jan K. S. – janks@puc-rio.br

ENG 1419 – Programação de Microcontroladores

# Revisão da Introdução

```
# código Python para verificar número primo
numero = int(input("Digite um número: "))

if numero > 1:
    for i in range(2, numero):
        if (numero % i) == 0:
            print(numero, "não é primo")
            break
    else:
        print(numero, "é primo")
else:
    print(numero, "não é primo")
```

```
// código C para verificar número primo
#include <stdio.h>

int main ()
{
    int numero, i;
    printf("Digite um número: ");
    scanf("%d", &numero);

    if (numero > 1) {
        for (i = 2; i < numero; i++) {
            if (numero % i == 0) {
                printf("%d não é primo", numero);
                break;
            }
            if (i == numero) {
                printf("%d é primo", numero);
            }
        } else {
            printf("%d não é primo", numero);
        }
    }
}
```

```
>>> x = 2  
  
>>> x  
2  
  
>>> y = x * 4 + 10  
  
>>> y  
18  
  
>>> y + 1  
19  
  
>>> y  
18  
  
>>> variavel_com_nome_grande = (x - y) / 8  
  
>>> variavel_com_nome_grande  
-1.875
```

```
>>> lista = [1, 2, 3]
>>> texto = "Texto"
>>> texto2 = 'Texto'
>>> dicionario = {'campo 1': 2, 'campo 2': 4}
>>> booleano = True
>>> booleano = False
>>> funcao = print
>>> vazio = None
>>> # Linha de comentário
```

```
>>> print('Olá!')  
Olá!  
  
>>> print('Valor: ', 2)  
Valor: 2  
  
>>> sleep(4)  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
NameError: name 'sleep' is not defined  
  
>>> from time import sleep ← importa função sleep da biblioteca time  
  
>>> sleep(4) ← pausa programa por 4 segundos  
  
>>> from time import * ← importa todas as funções da biblioteca time
```

```
>>> lista = [10, 20, 30]           ← lista de 3 elementos
>>> lista[0]                      ← acesso ao 1º elemento (índice 0)
10

>>> lista[1]
20

>>> lista[2]
30

>>> lista[3]                      ← 4º elemento (índice 3) não existe
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range

>>> lista2 = ['abc', [0, 0], None] ← elementos de tipos diferentes
>>> lista + lista2                ← concatenação de duas listas
[10, 20, 30, 'abc', [0, 0], None]
```

```
>>> lista = [10, 20, 30]
>>> lista.append(5)                                     ← adiciona o elemento 5 no final
>>> lista
[10, 20, 30, 5]
>>> lista.remove(2)                                    ← remove o elemento no índice 2
>>> lista
[10, 20, 5]
>>> lista.sort()                                      ← ordena lista
>>> lista
[5, 10, 20]
>>> lista.sort(reverse=True)                          ← parâmetro opcional para usar
                                                        ordem decrescente
>>> lista
[20, 10, 5]
```

```
>>> dados = {'nome': 'Jan K. S.', 'idade': 31}  
  
>>> dados['nome']                                ← acessa um valor do dicionário  
'Jan K. S.'  
  
>>> dados['idade'] = 32                         ← altera um valor do dicionário  
  
>>> dados['sexo'] = 'masculino'                 ← adiciona um valor no dicionário  
  
>>> del dados['nome']                           ← remove um valor do dicionário  
  
>>> dados  
{'idade': 32, 'sexo': 'masculino'}
```

```
>>> texto = "This is Sparta!"  
>>> texto = 'This is Sparta!'  
>>> texto[2]                                ← caracter no índice 2 = 3º caracter  
i  
>>> texto.replace('is', 'iiiiisssss')  
>>> texto  
Thiiiiisssss iiiiiisssss Sparta!  
>>> 'Madness? ' + texto  
Madness? Thiiiiisssss iiiiiisssss Sparta!  
>>> 'Madness? ' + '\n' + texto ← '\n' representa uma quebra de linha  
Madness?  
Thiiiiisssss iiiiiisssss Sparta!
```

```
>>> x = 2                                     ← atribui x ao valor 2
>>> x == 2                                    ← verifica se x é igual a 2
True
>>> x != 2                                    ← verifica se x é diferente de 2
False
>>> x > 3                                     
False
>>> not x <= 0
True
>>> x > 0 and x < 1
False
>>> x <= -2 or x >= 2
True
>>> x in [1, 3, 5, 7]                         ← verifica se x está dentro da lista
False
>>> 'abc' in {'abc': 1, 'cde': 10}           ← verifica se o dicionário
True                                         contém a chave
```

```
>>> x = 2                                     ← "se" + verificação
>>> if x < 0:
...     y = 4
... elif x > 0 and x < 7:                     ← “caso contrário, se” (else if)
...     y = 3
... else:                                       ← “caso contrário”
...     y = 0
...
...
>>> y
3
```

```
>>> for i in [1, 2, 3, 4]:  
...     y = i * 2  
...     print(y)
```

repete comandos para cada elemento "i" da lista

```
...  
2  
4  
6  
8
```

```
>>> for i in range(1, 4):
```

intervalo de 1 até ANTES de 4

```
...     print(i)  
...     sleep(1)
```

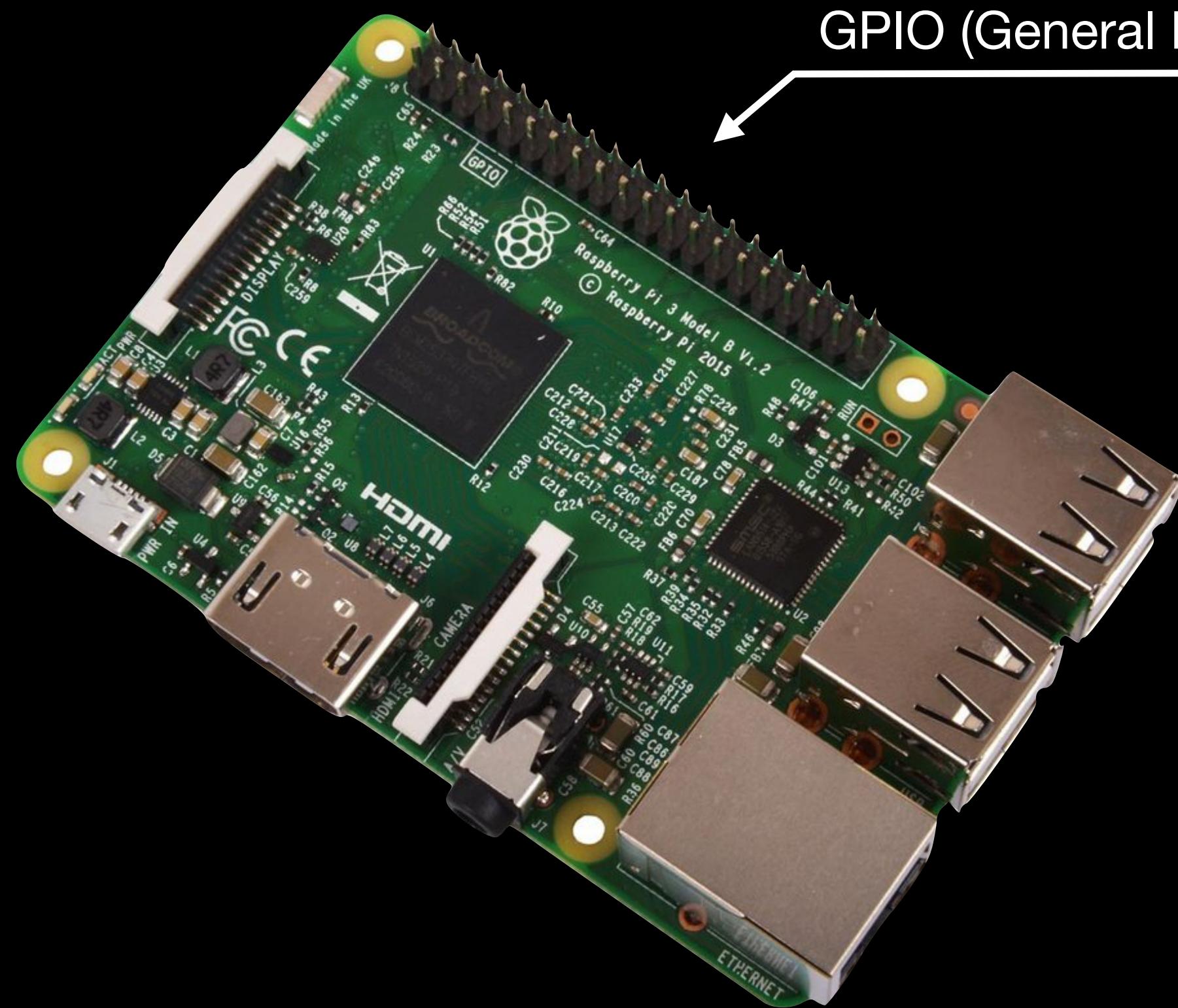
espera 1 segundo entre cada repetição

```
...  
1  
2  
3
```

```
>>> x = 15  
  
>>> while x > 1:           ← repita enquanto x for maior que 5  
...     x = x / 5  
  
...  
  
>>> x  
0.6  
  
>>> while True:           ← repita PARA SEMPREEEEEEE  
...     print('Hey, listen!')  
...     sleep(0.5)  
  
...  
  
Hey, listen!  
Hey, listen!  
Hey, listen!  
  
...
```

```
>>> def faz_conta(x, y):  
...     return (x + y) / (x * y)  
...  
>>> def imprime_resultado(x):  
...     print('O resultado obtido foi:', x)  
...  
>>> faz_conta(2, 4)  
0.75  
>>> imprime_resultado( faz_conta(2, 4) )  
O resultado obtido foi: 0.75
```

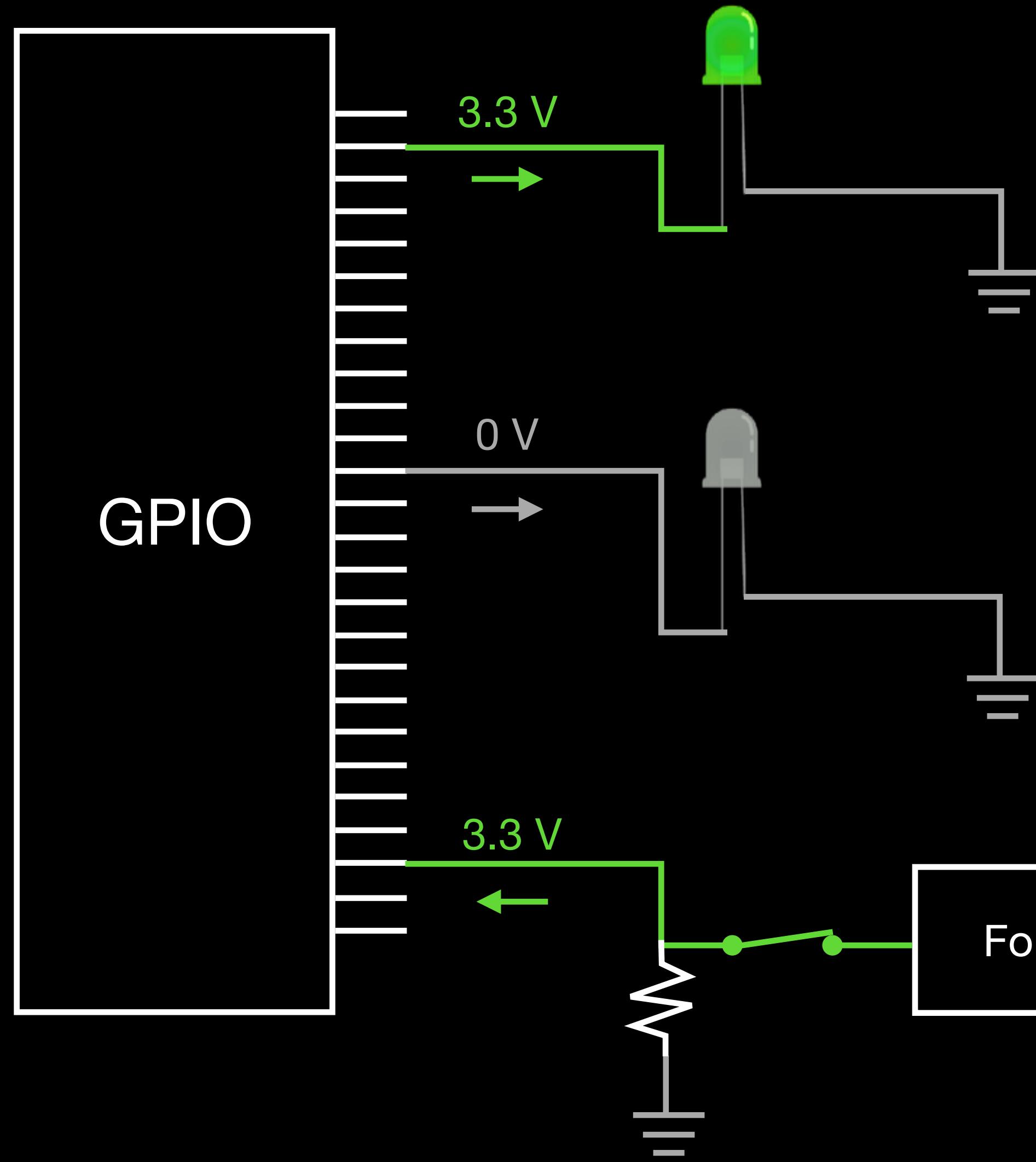
# Hardware



GPIO (General Purpose Input Output)

GPIO

```
>>> ligue a porta 2
```



```
>>> desligue a porta 12
```

```
>>> leia a porta 25
```

Controle Digital Através da GPIO

3.3 V



nivel lógico 1  
**HIGH**

0 V

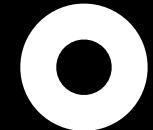


nivel lógico 0  
**LOW**

Texto do Título

# Raspberry Pi 3 GPIO Header

| Pin# | NAME                               | NAME                               | Pin# |
|------|------------------------------------|------------------------------------|------|
| 01   | 3.3v DC Power                      | DC Power 5v                        | 02   |
| 03   | GPIO02 (SDA1 , I <sup>2</sup> C)   | DC Power 5v                        | 04   |
| 05   | GPIO03 (SCL1 , I <sup>2</sup> C)   | Ground                             | 06   |
| 07   | GPIO04 (GPIO_GCLK)                 | (TXD0) GPIO14                      | 08   |
| 09   | Ground                             | (RXD0) GPIO15                      | 10   |
| 11   | GPIO17 (GPIO_GEN0)                 | (GPIO_GEN1) GPIO18                 | 12   |
| 13   | GPIO27 (GPIO_GEN2)                 | Ground                             | 14   |
| 15   | GPIO22 (GPIO_GEN3)                 | (GPIO_GEN4) GPIO23                 | 16   |
| 17   | 3.3v DC Power                      | (GPIO_GEN5) GPIO24                 | 18   |
| 19   | GPIO10 (SPI_MOSI)                  | Ground                             | 20   |
| 21   | GPIO09 (SPI_MISO)                  | (GPIO_GEN6) GPIO25                 | 22   |
| 23   | GPIO11 (SPI_CLK)                   | (SPI_CE0_N) GPIO08                 | 24   |
| 25   | Ground                             | (SPI_CE1_N) GPIO07                 | 26   |
| 27   | ID_SD (I <sup>2</sup> C ID EEPROM) | (I <sup>2</sup> C ID EEPROM) ID_SC | 28   |
| 29   | GPIO05                             | Ground                             | 30   |
| 31   | GPIO06                             | GPIO12                             | 32   |
| 33   | GPIO13                             | Ground                             | 34   |
| 35   | GPIO19                             | GPIO16                             | 36   |
| 37   | GPIO26                             | GPIO20                             | 38   |
| 39   | Ground                             | GPIO21                             | 40   |

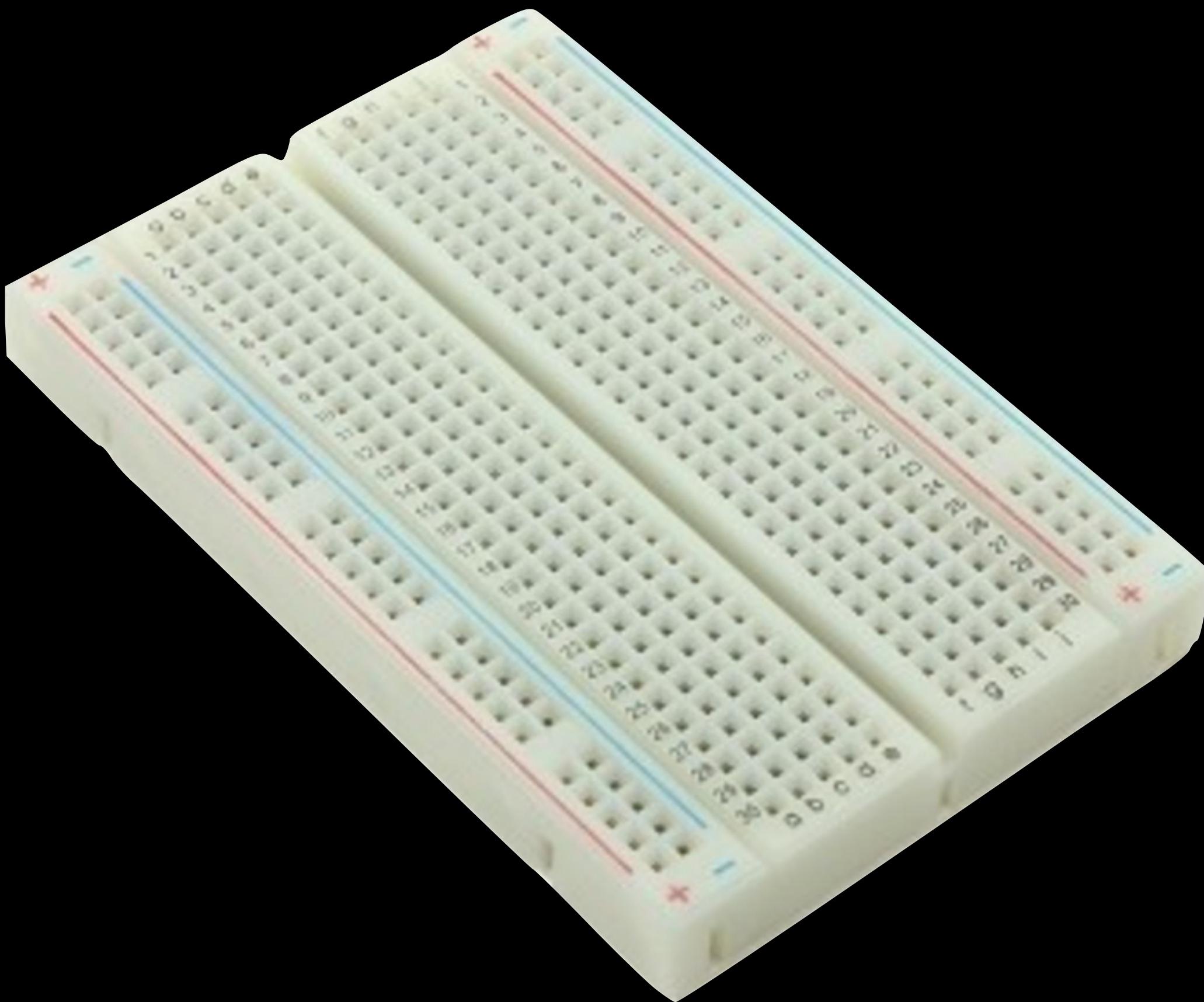
 Porta Reservada (Indisponível)

 Porta Disponível

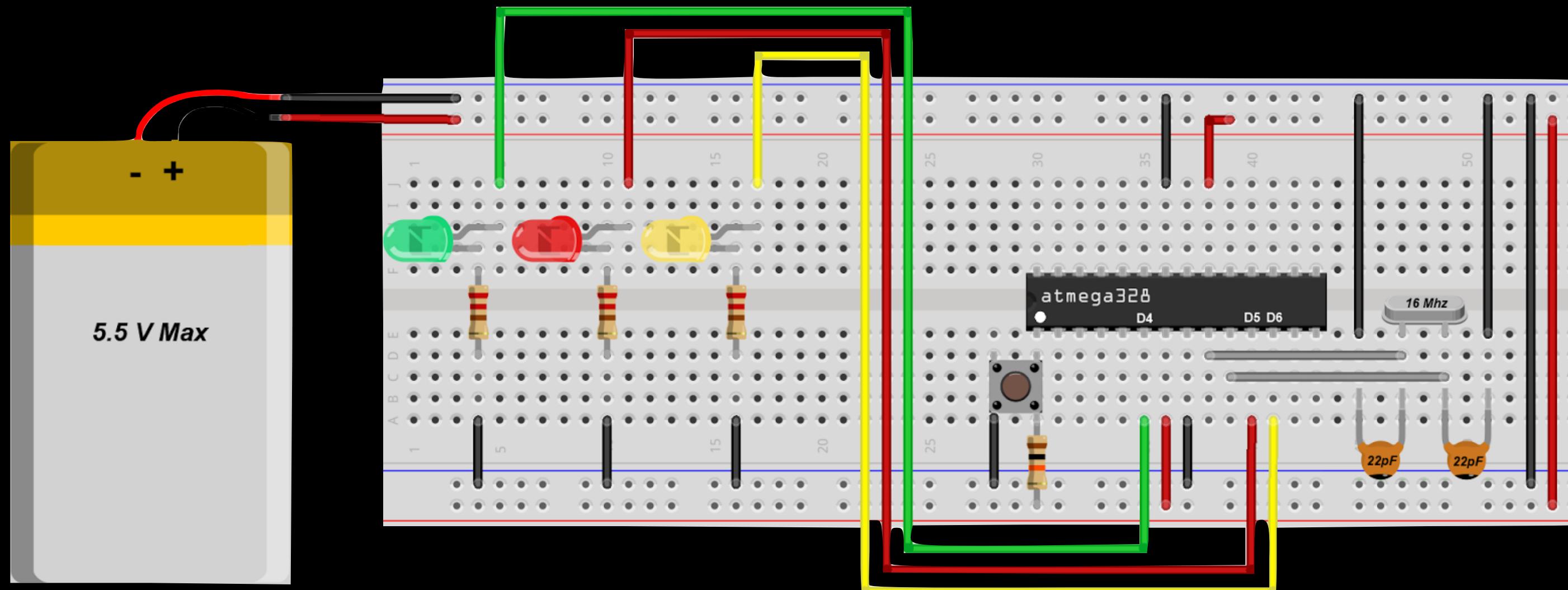
 Porta Disponível com Propriedades Extras

↓↓ (orientação com as entradas USBs para baixo) ↓↓

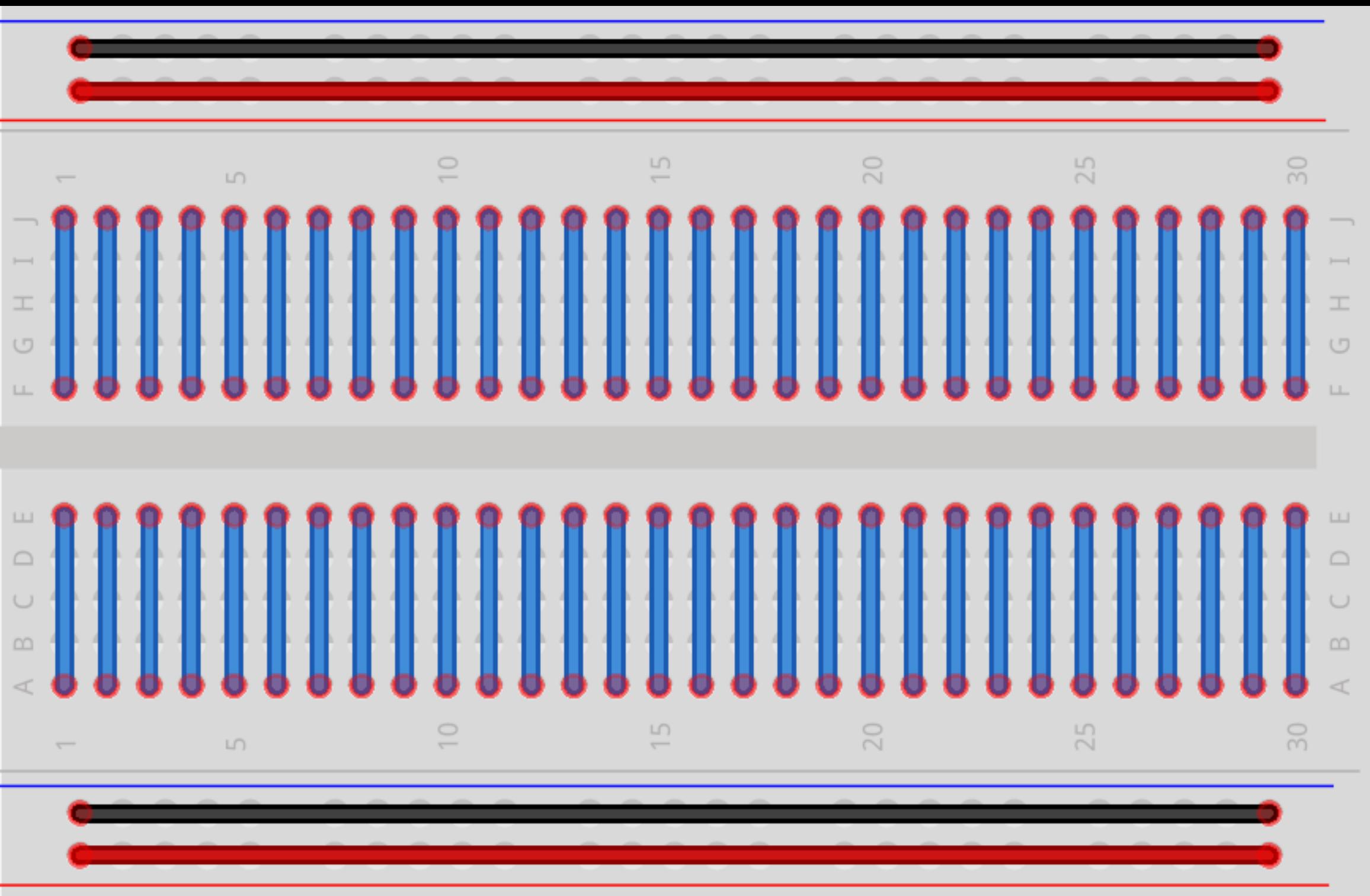
Identificação das Portas da GPIO



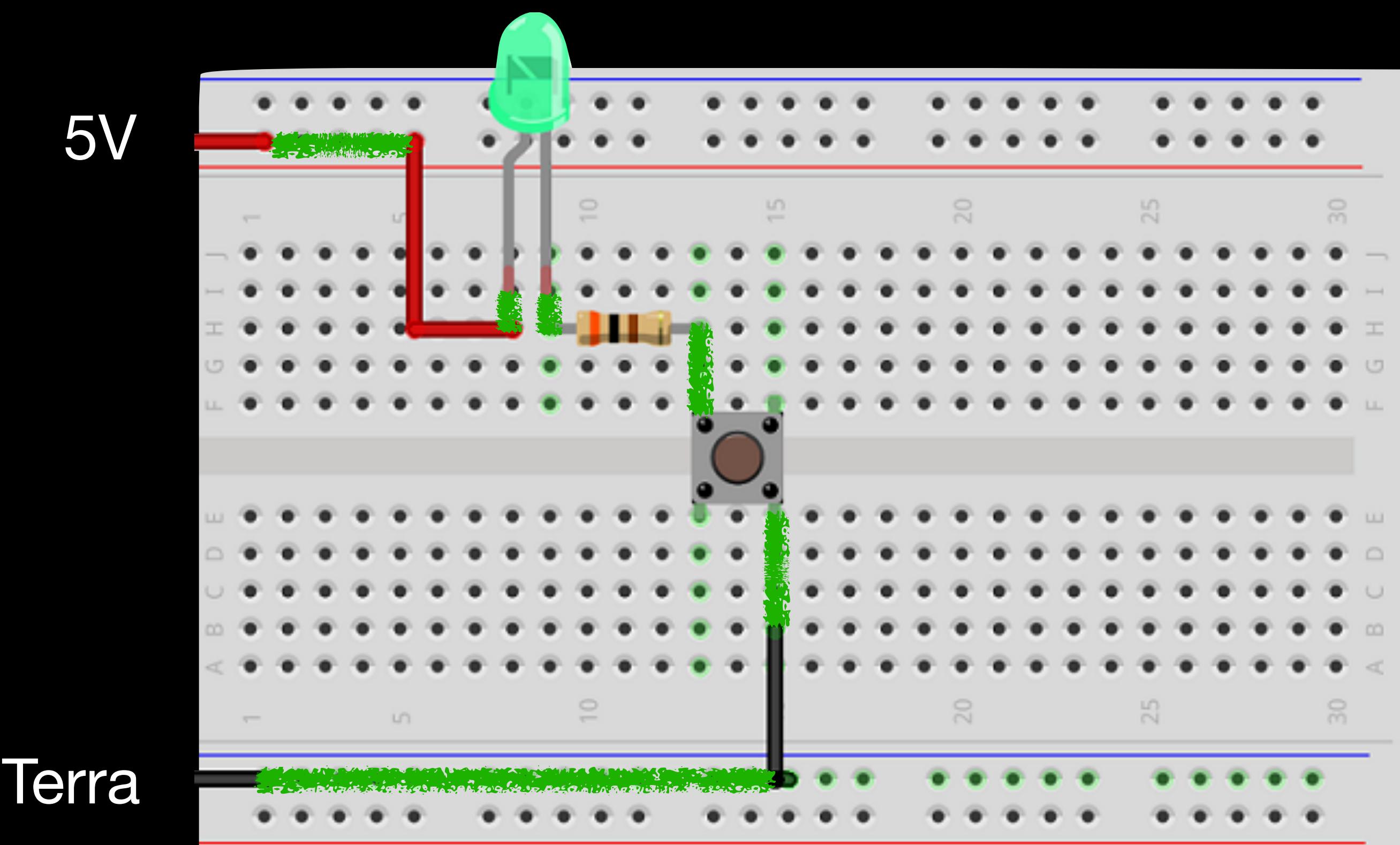
Protopboard



Componentes Eletrônicos em uma Protoboard



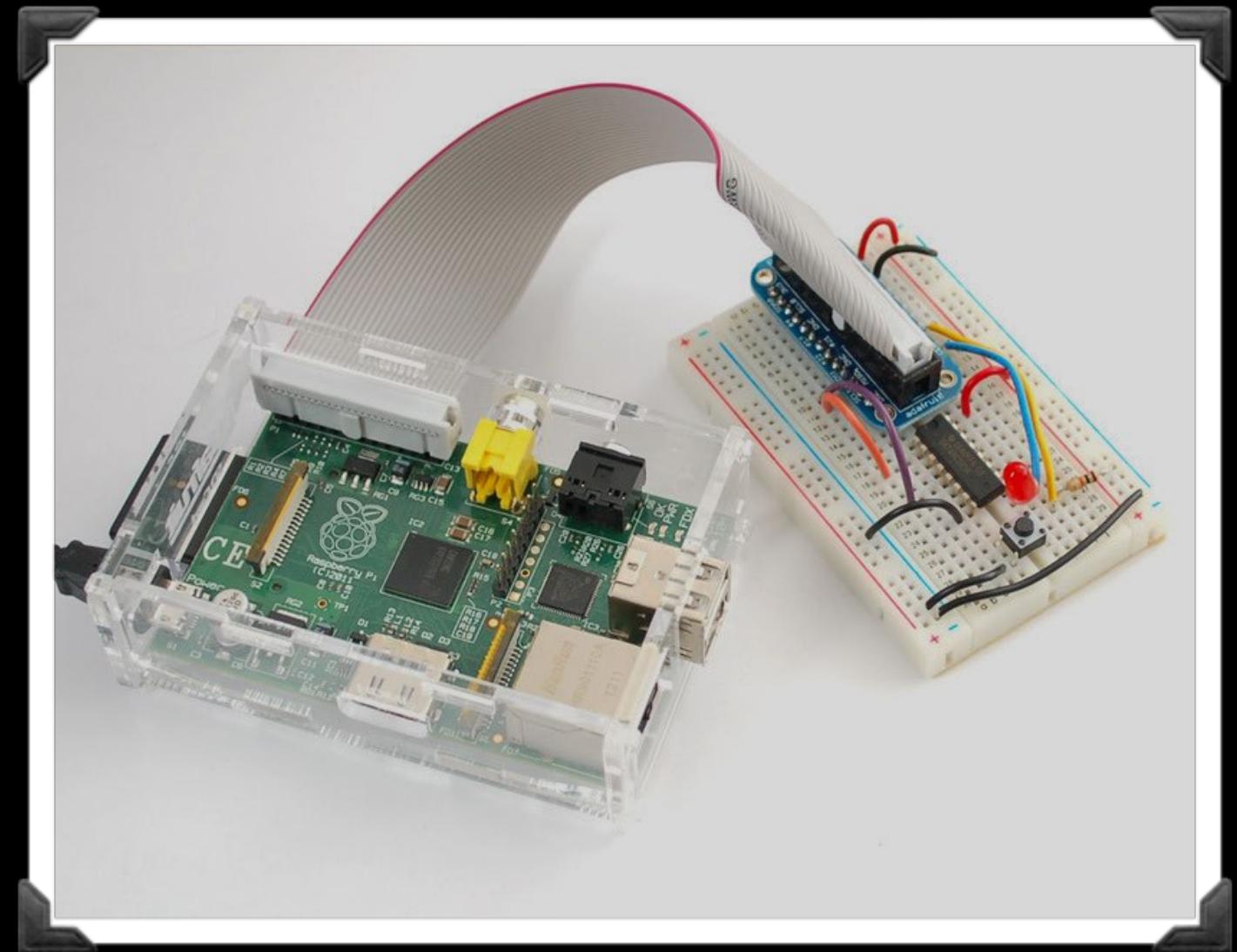
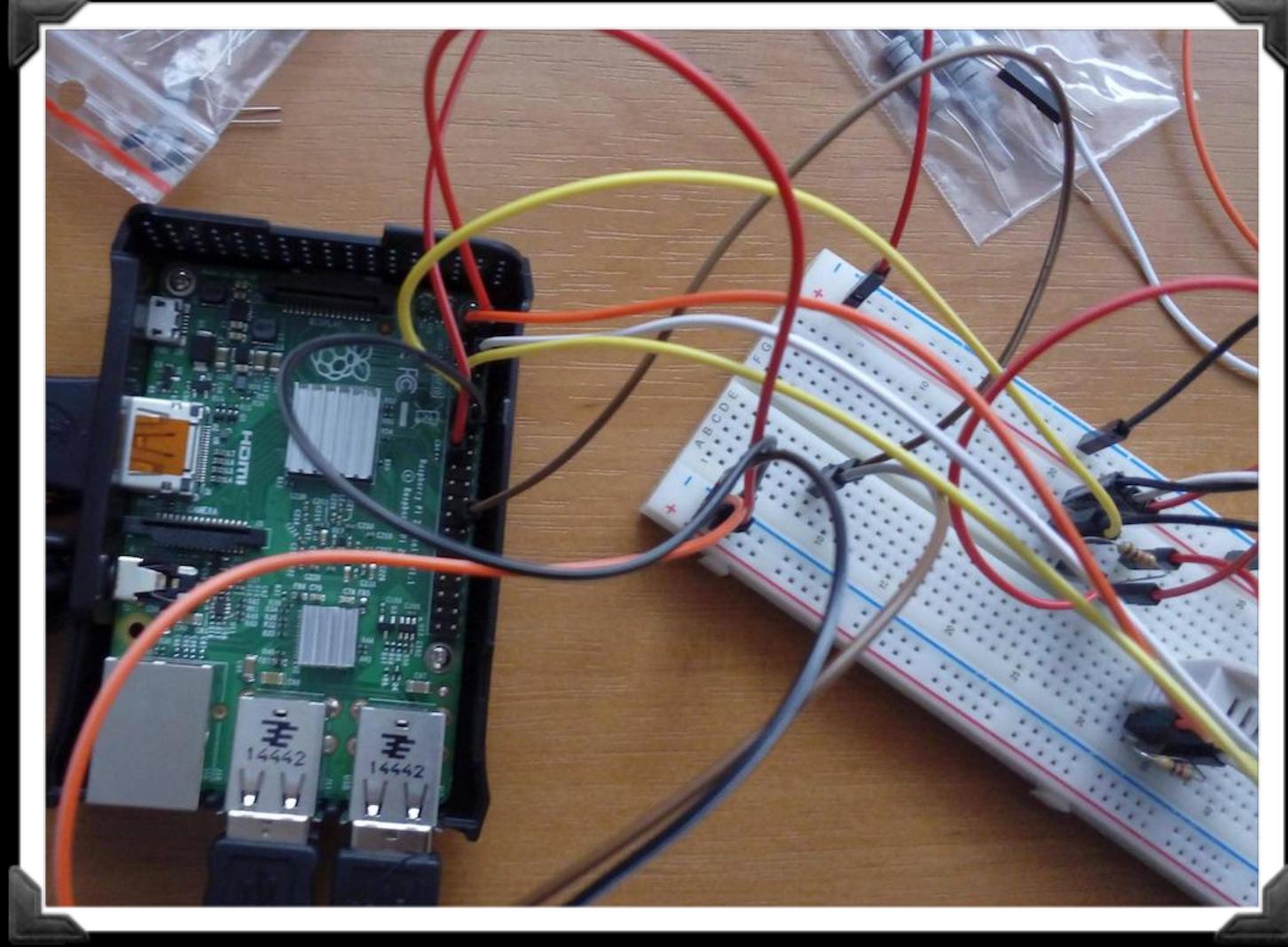
Conexões Internas da Protoboard



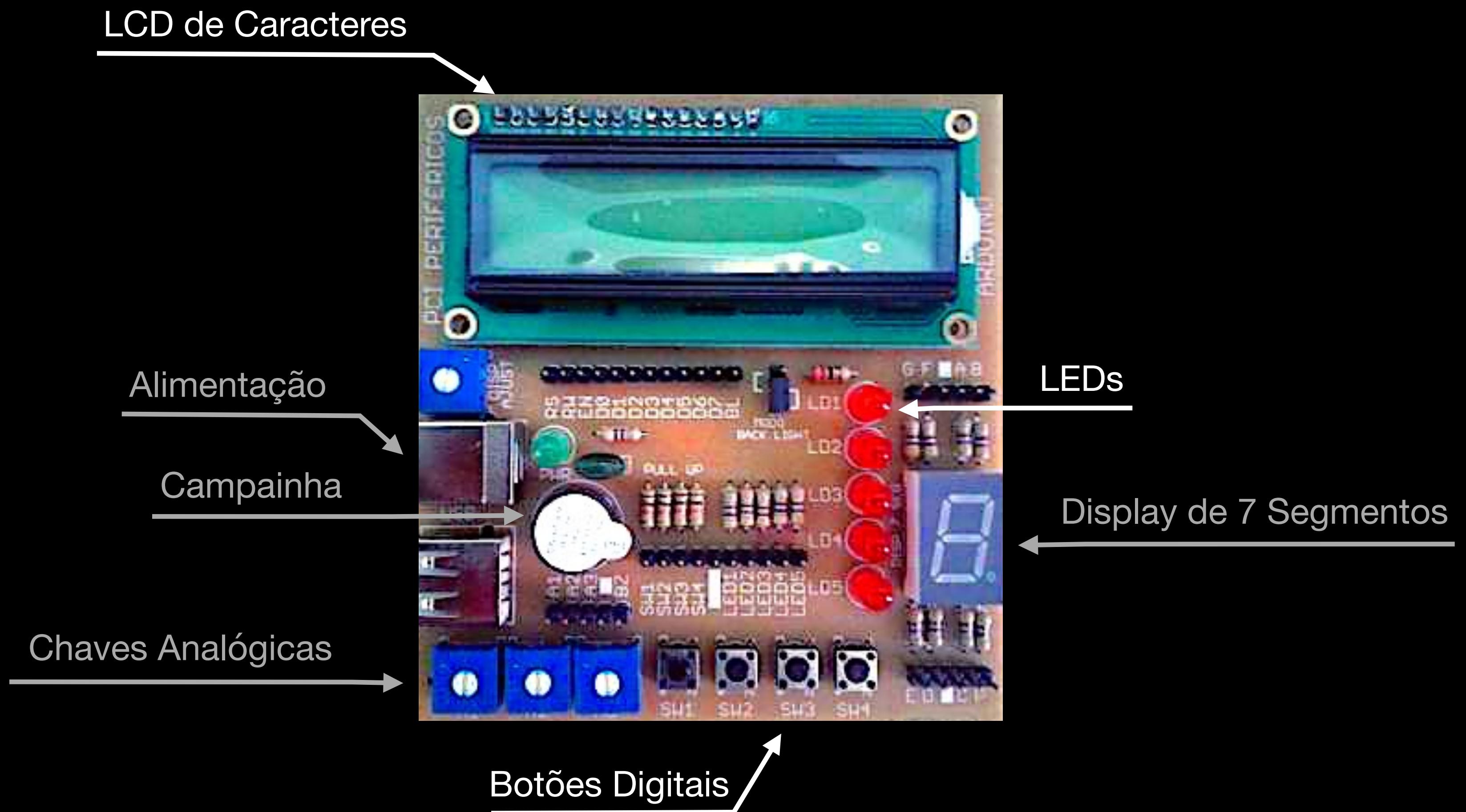
Componentes Conectados Internamente



Cabo "Flat" com Identificação

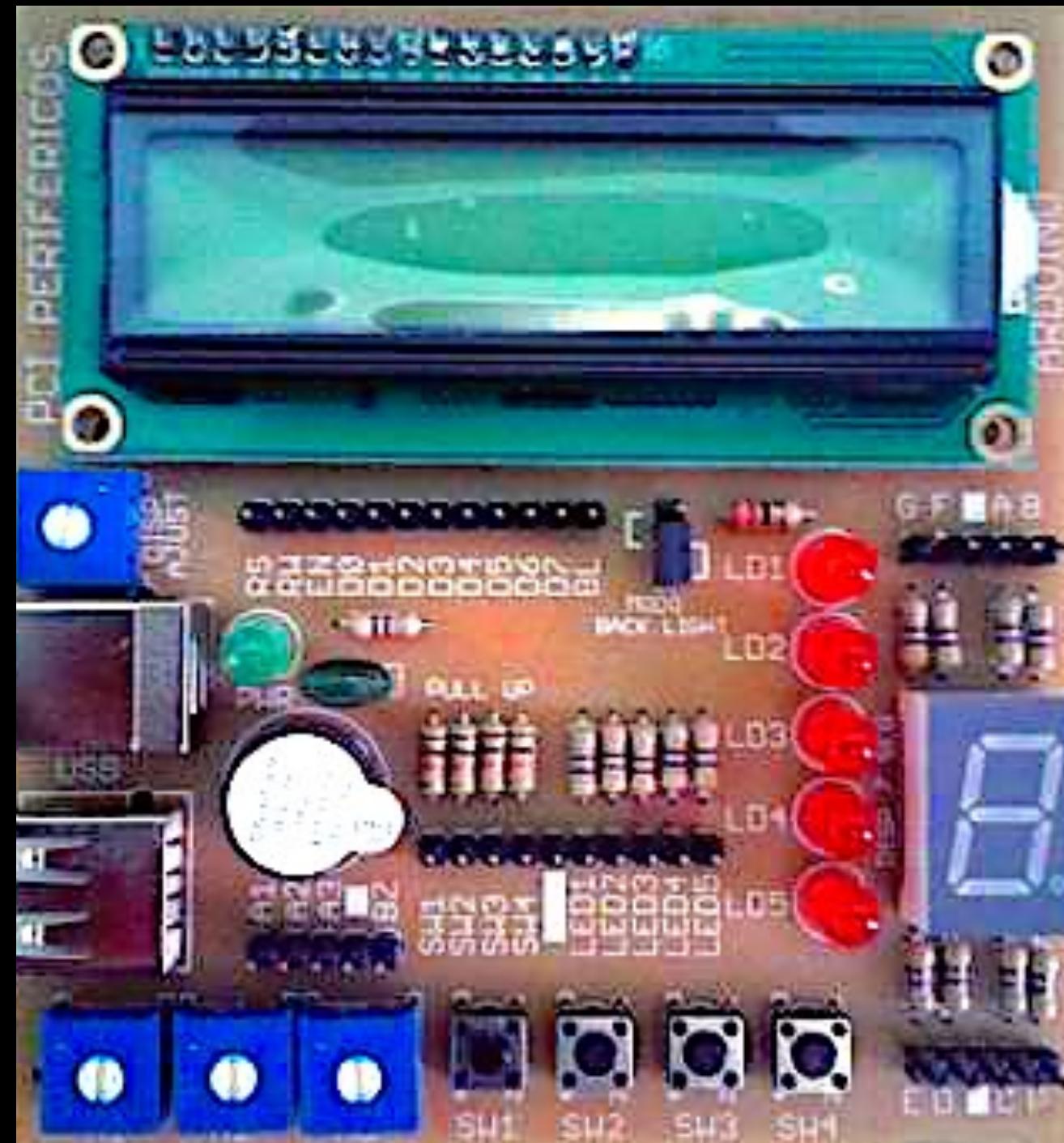


Separação entre Microcontrolador e Circuito



Placa de Periféricos Básicos

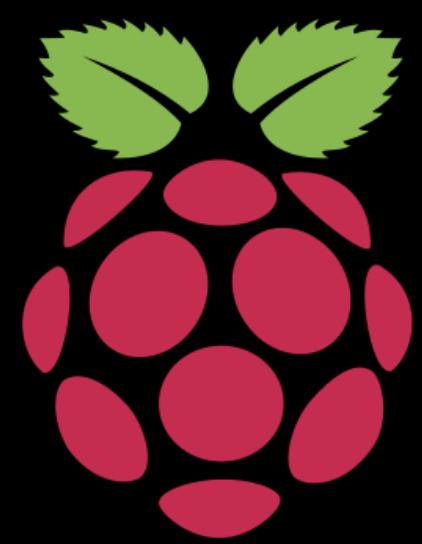
GPIO 2, 3, 4, 5, 6, 7



GPIO 21  
GPIO 22  
GPIO 23  
GPIO 24  
GPIO 25

GPIO 11, 12, 13, 14

Conexões dos Botões, LEDs e LCD de Caracters com as Portas da GPIO

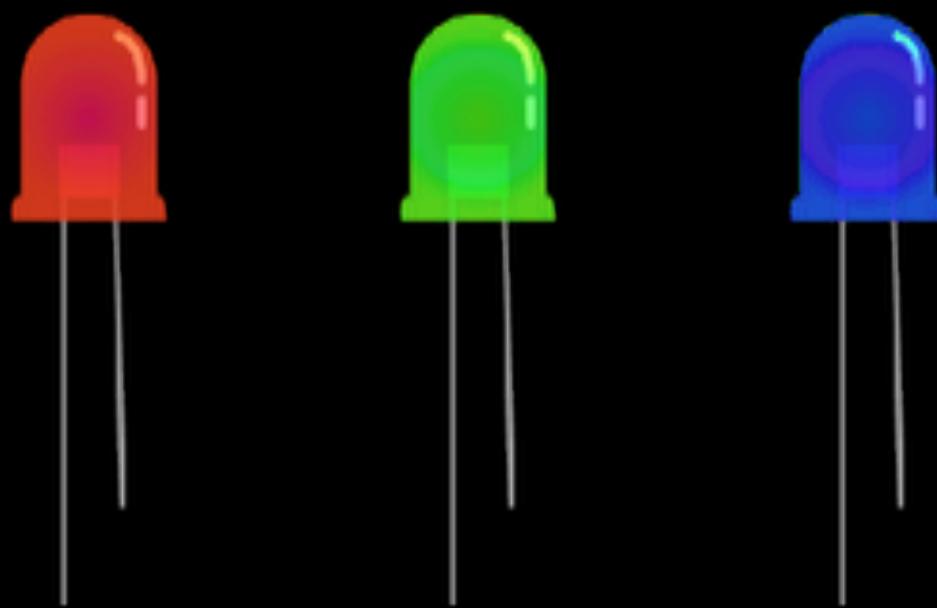


gpiozero

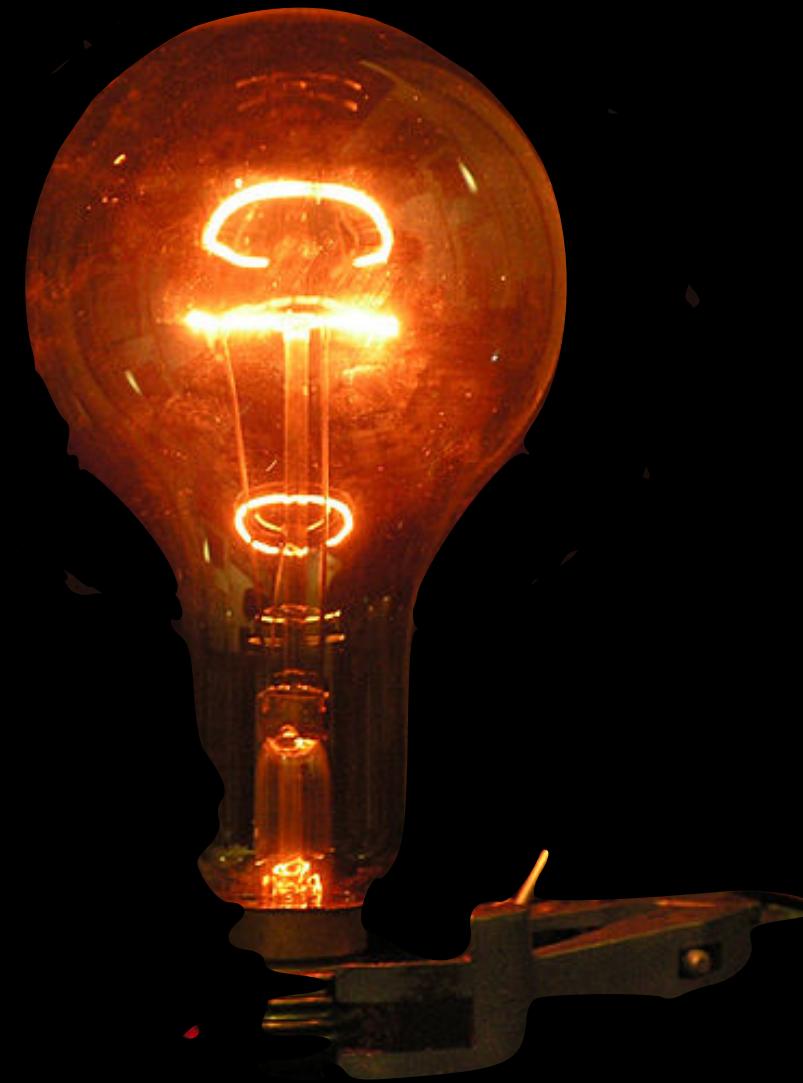


Adafruit\_Python\_CharLCD

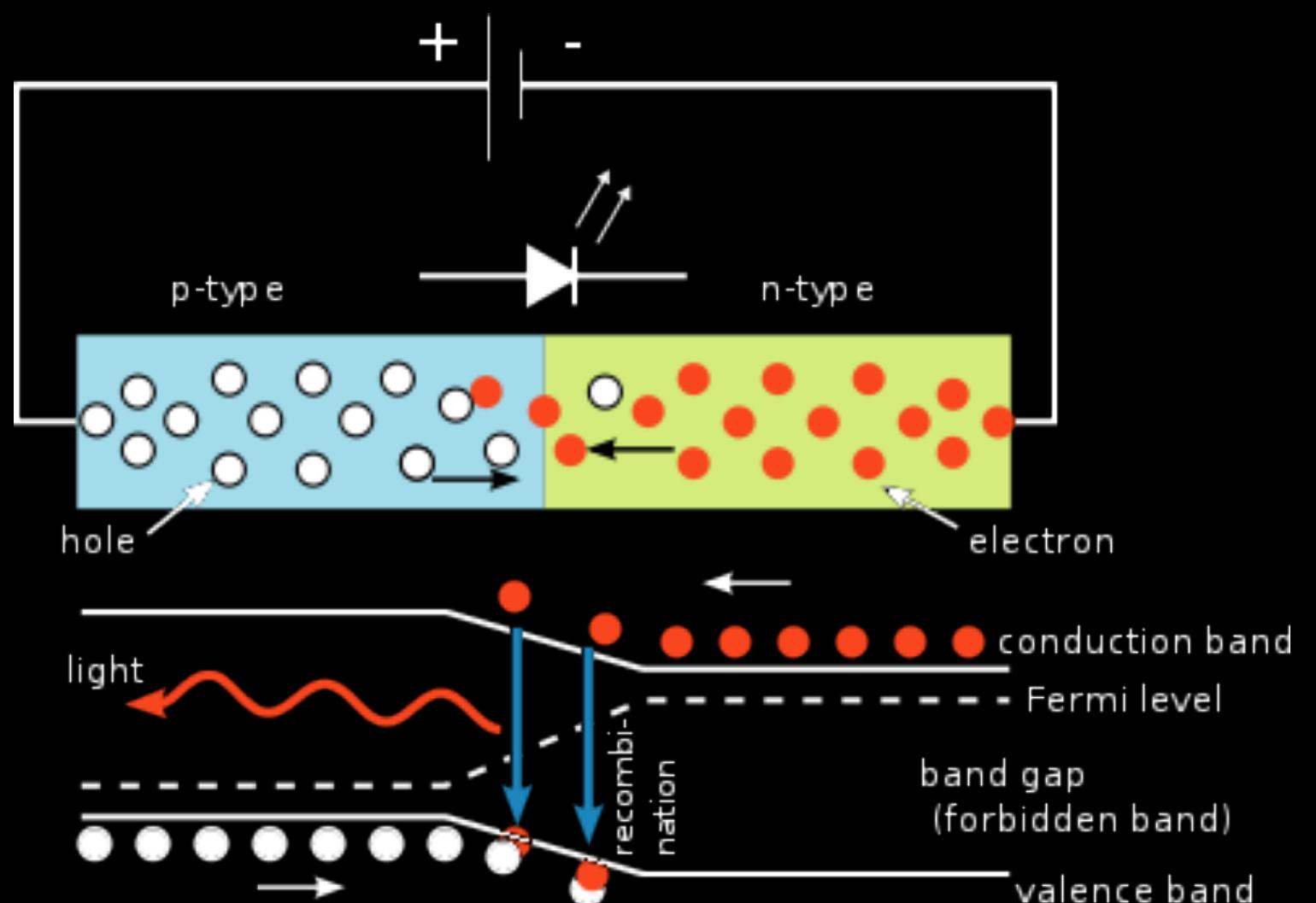
Bibliotecas para Manipular Periféricos Básicos



LEDs



luz emitida pelo calor



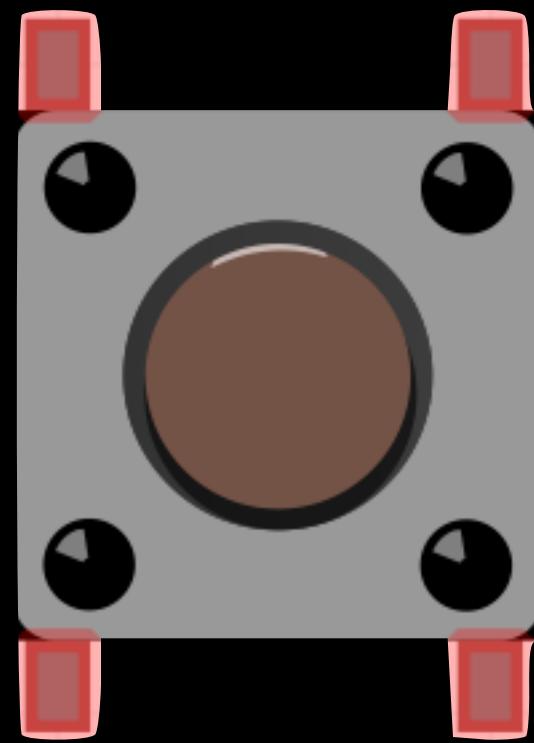
luz emitida por elétrons no silício

Lâmpadas Incandescentes vs LEDs

```
>>> from gpiozero import LED  
  
>>> led = LED(21)           ← configura a porta 21 da GPIO como um LED  
  
>>> led.on()                ← acende o LED  
  
>>> led.is_lit              ← verifica se o LED está aceso  
True  
  
>>> led.off()  
  
>>> led.is_lit              ← verifica se o LED está aceso  
False  
  
>>> led.toggle()            ← alterna o estado atual entre aceso e apagado  
  
>>> led.is_lit              ← verifica se o LED está aceso  
True  
  
>>> led.toggle()  
  
>>> led.is_lit              ← verifica se o LED está aceso  
False
```

```
>>> from gpiozero import LED  
>>> led = LED(22)  
>>> led.blink() ← pisque continuamente  
>>> led.blink(on_time=0.5, off_time=2.0) ← define tempo  
      aceso e apagado  
>>> led.blink(n=4) ← pisque apenas 4 vezes  
>>> led.blink(n=4, background=False) ← aguarde parar de  
      piscar para prosseguir
```

Função para Piscar o LED



Botão

```
>>> from gpiozero import Button  
>>> botao = Button(11) ← configura a porta 11 da GPIO como um botão  
>>> botao.is_pressed ← verifica se o botão está pressionado no momento  
False  
>>> botao.wait_for_press() ← pausa o programa até que o botão seja  
pressionado
```

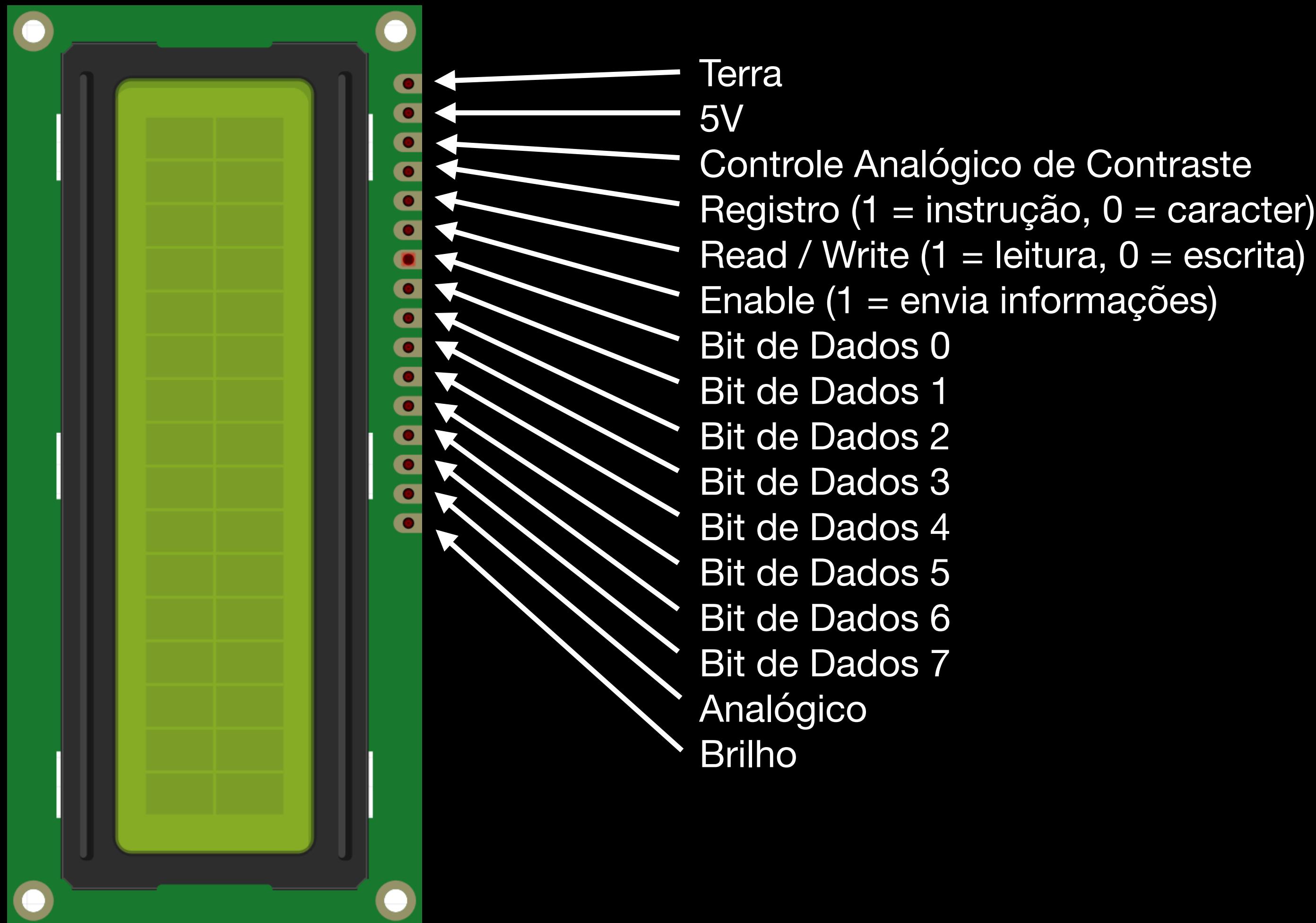
```
>>> def minha_funcao():
...     print("Lançar míssel nuclear!")
...
...
>>> from gpiozero import Button, LED
>>> botao1 = Button(11)                                chame minha_funcao ao pressionar
>>> botao1.when_pressed = minha_funcao                (Obs: não coloque parênteses no final)
>>> botao2 = Button(13)
>>> botao2.when_released = minha_funcao
>>> botao3 = Button(12)
>>> led = LED(22)                                     chame toggle do LED ao pressionar
>>> botao3.when_pressed = led.toggle
```



LCD de Caracteres



Vídeo sobre o Display LCD 16 x 2 no YouTube



Pinos do LCD de Caracteres

| Instruction                 | Instruction code |     |     |     |     |     |     |     |     |     |  | Description   | Execution time<br>(fosc=270KHz) |
|-----------------------------|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|---|---------------------------------|
|                             | RS               | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |  |   |                                 |
| Clear Display               | 0                | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   |  | Clears entire display and sets DDRAM address to 00H.  | 1.53ms                          |
| Return Home                 | 0                | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | -   |  | Sets DDRAM address to 00H in AC and returns shifted display to its original position. The contents of DDRAM remain unchanged.         | 1.53ms                          |
| Entry Mode Set              | 0                | 0   | 0   | 0   | 0   | 0   | 0   | 1   | I/D | SH  |  | Sets cursor move direction and enable the shift of entire display. These operations are performed during data write and read.         | 39 μ s                          |
| Display ON/OFF Control      | 0                | 0   | 0   | 0   | 0   | 0   | 1   | D   | C   | B   |  | Set ON/OFF of entire display (D), cursor ON/OFF(C), and blinking of cursor position character(B).                                     | 39 μ s                          |
| Cursor or Display Shift     | 0                | 0   | 0   | 0   | 0   | 1   | S/C | R/L | -   | -   |  | Moves cursor and shifts display without changing DDRAM contents.  | 39 μ s                          |
| Function Set                | 0                | 0   | 0   | 0   | 1   | DL  | N   | F   | -   | -   |  | Sets interface data length (DL: 8-bit/4-bit), numbers of display line (N: 2-line/1-line), and display font type (F: 5x11dots/5x8dots) | 39 μ s                          |
| Set CGRAM Address           | 0                | 0   | 0   | 1   | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 |  | Set CGRAM address in address counter.   | 39 μ s                          |
| Set DDRAM Address           | 0                | 0   | 1   | AC6 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 |  | Set DDRAM address in address Counter.   | 39 μ s                          |
| Read Busy Flag and Address  | 0                | 1   | BF  | AC6 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 |  | Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.                             | 0 μ s                           |
| Write data to CG or DD RAM  | 1                | 0   | D7  | D6  | D5  | D4  | D3  | D2  | D1  | D0  |  | Write data into internal RAM (DDRAM/CGRAM).   | 43us                            |
| Read data from CG or DD RAM | 1                | 1   | D7  | D6  | D5  | D4  | D3  | D2  | D1  | D0  |  | Read data from internal RAM (DDRAM/CGRAM).  | 43us                            |

| Upper 4 bits<br>Lower 4 bits | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|------------------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| CG RAM (1)                   | 0000 | CG   | P    | ^P   |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (2)                   | 0001 | CG   | !1A  | Qa9  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (3)                   | 0010 | CG   | "2B  | Rbr  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (4)                   | 0011 | CG   | #3C  | Scs  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (5)                   | 0100 | CG   | \$4D | Tdt  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (6)                   | 0101 | CG   | %5E  | Ueu  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (7)                   | 0110 | CG   | &6F  | Vfv  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (8)                   | 0111 | CG   | ?7G  | WgW  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (1)                   | 1000 | CG   | (8H  | Xhx  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (2)                   | 1001 | CG   | )9I  | Yiy  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (3)                   | 1010 | CG   | *:J  | Zjz  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (4)                   | 1011 | CG   | +;K  | Ck{  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (5)                   | 1100 | CG   | ,<L  | ¥1I  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (6)                   | 1101 | CG   | --=M | ]m}  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (7)                   | 1110 | CG   | .>N  | ^n→  |      |      |      |      |      |      |      |      |      |      |      |      |
| CG RAM (8)                   | 1111 | CG   | /?O  | _o←  |      |      |      |      |      |      |      |      |      |      |      |      |

Comandos para o LCD

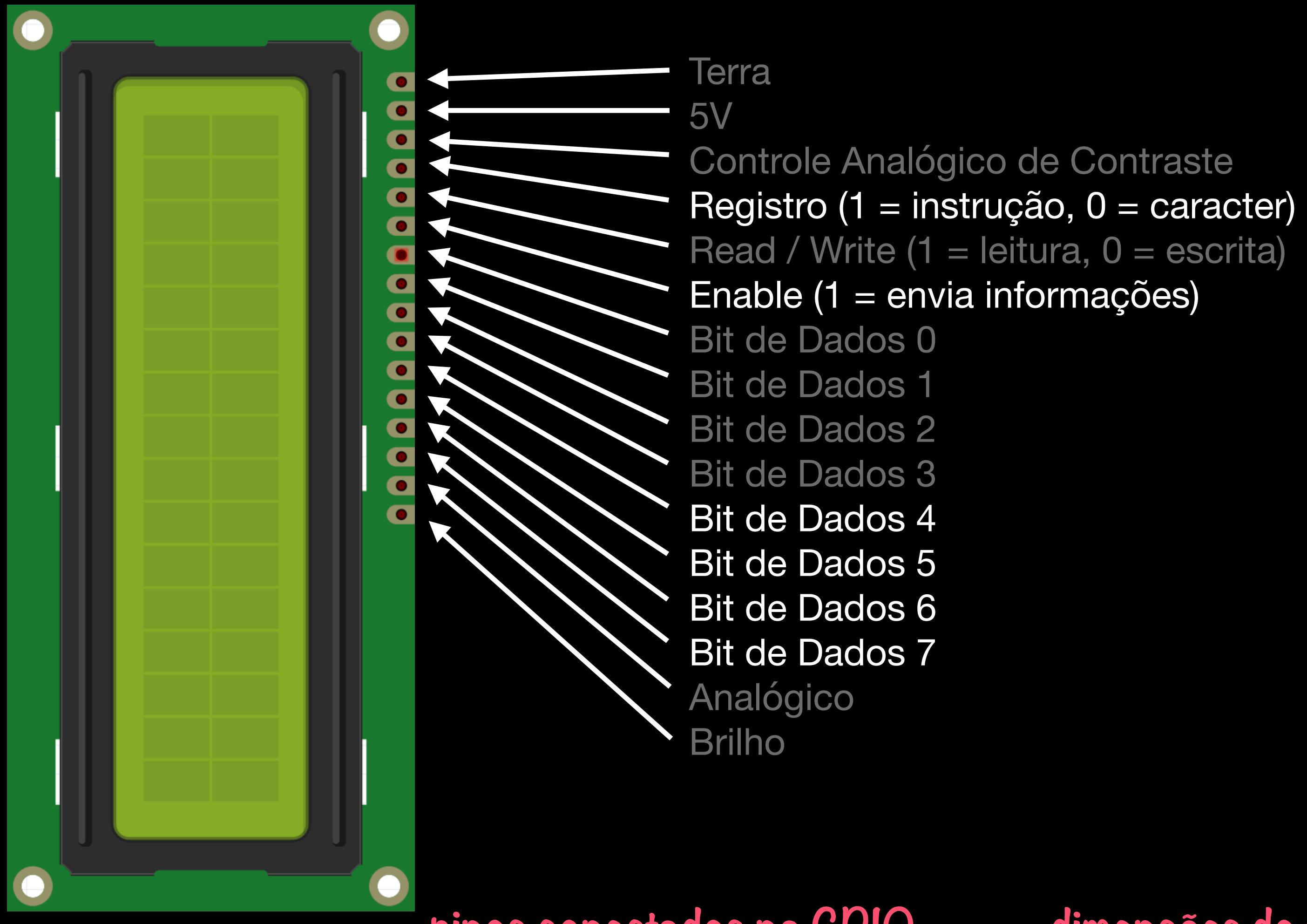


Precisamos implementar  
todos esses comandos e  
códigos???

Relaxa, a gente já fez tudo  
isso para você!



Comandos para o LCD



*pinos conectados na GPIO*

*dimensões do LCD*

Adafruit\_CharLCD(RS, EN, D4, D5, D6, D7, colunas, linhas)

Pinos Usados pela Biblioteca da Adafruit

```
Adafruit_CharLCD(RS, EN, D4, D5, D6, D7, colunas, linhas)
```



no nosso caso...

```
Adafruit_CharLCD(2, 3, 4, 5, 6, 7, 16, 2)
```

Pinos Usados pela Biblioteca da Adafruit

```
>>> from Adafruit_CharLCD import Adafruit_CharLCD  
>>> lcd = Adafruit_CharLCD(2, 3, 4, 5, 6, 7, 16, 2)  
>>> lcd.message("Teste")  
>>> lcd.message("Teste 2")  
>>> lcd.clear()  
>>> lcd.message("Teste 3")
```



Exibição de Mensagens no LCD de Caracteres

```
>>> from Adafruit_CharLCD import Adafruit_CharLCD  
>>> lcd = Adafruit_CharLCD(2, 3, 4, 5, 6, 7, 16, 2)  
>>> lcd.message("Teste com texto longo")  
>>> lcd.clear()  
>>> lcd.message("Teste com\n texto longo")
```



*importação de bibliotecas*

*definição de funções auxiliares*

*inicialização de variáveis*

`while True:`

`...`

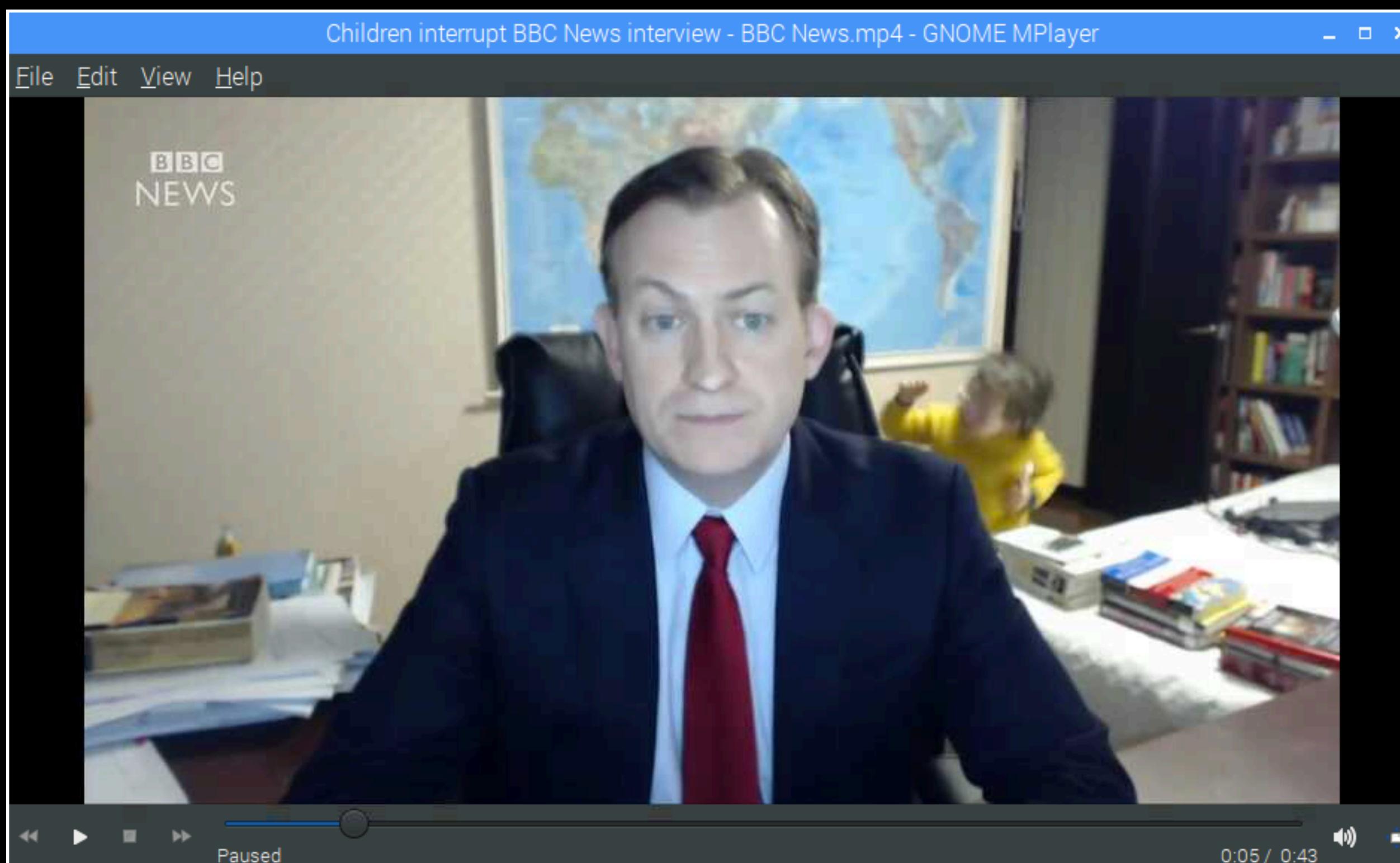
`sleep`

# Software

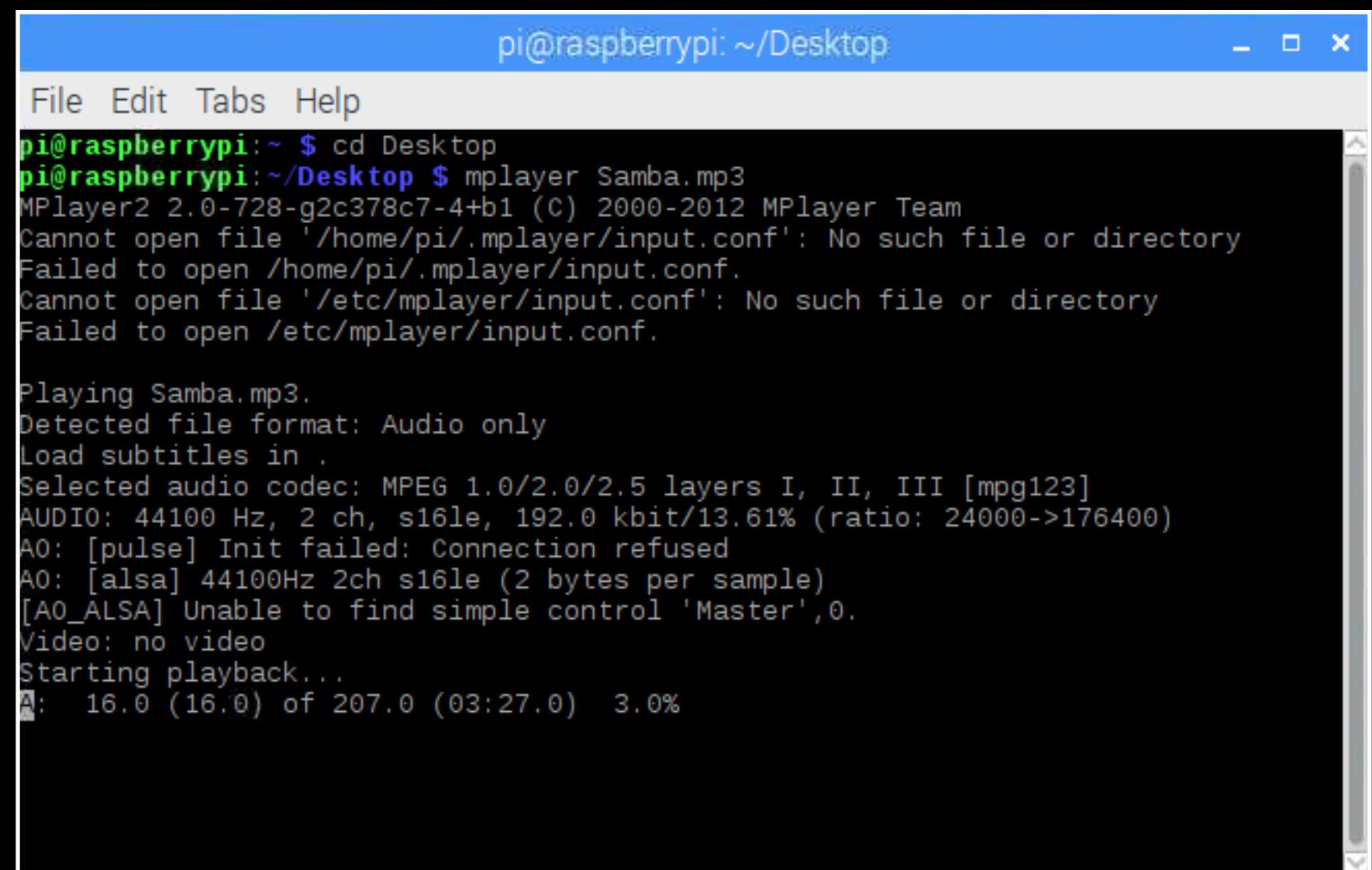
**M**P*layer*



Texto do Título



Interface Gráfica do MPlayer



A screenshot of a terminal window titled "pi@raspberrypi: ~/Desktop". The window has a blue header bar with standard window controls (minimize, maximize, close) on the right. Below the title bar is a menu bar with "File", "Edit", "Tabs", and "Help". The main area of the terminal shows the command-line interface of MPlayer. The user has run the command "mplayer Samba.mp3". The output indicates that MPlayer is unable to find configuration files in the expected locations (home/pi/.mplayer/input.conf and /etc/mplayer/input.conf). It then proceeds to play the file, showing details about the audio codec (MPEG 1.0/2.0/2.5 layers I, II, III [mpg123]), sample rate (44100 Hz), channels (2 ch), and bit rate (192.0 kbit). It also mentions that audio output (A0) failed due to a connection refused and switched to ALSA. Video playback is noted as failing to find a simple control 'Master'. The playback progress shows 16.0 seconds of a 207.0-second file at 3.0% completion.

```
pi@raspberrypi: ~/Desktop
pi@raspberrypi:~/Desktop $ mplayer Samba.mp3
MPlayer2 2.0-728-g2c378c7-4+b1 (C) 2000-2012 MPlayer Team
Cannot open file '/home/pi/.mplayer/input.conf': No such file or directory
Failed to open /home/pi/.mplayer/input.conf.
Cannot open file '/etc/mplayer/input.conf': No such file or directory
Failed to open /etc/mplayer/input.conf.

Playing Samba.mp3.
Detected file format: Audio only
Load subtitles in .
Selected audio codec: MPEG 1.0/2.0/2.5 layers I, II, III [mpg123]
AUDIO: 44100 Hz, 2 ch, s16le, 192.0 kbit/13.61% (ratio: 24000->176400)
A0: [pulse] Init failed: Connection refused
A0: [alsa] 44100Hz 2ch s16le (2 bytes per sample)
[A0_ALSA] Unable to find simple control 'Master', 0.
Video: no video
Starting playback...
A: 16.0 (16.0) of 207.0 (03:27.0) 3.0%
```

Uso do MPlayer via Terminal

The screenshot shows a GitHub repository page for 'baudm / mplayer.py'. The page title is 'baudm/mplayer.py: Lightweight Python wrapper for MPlayer'. The repository description is 'Lightweight Python wrapper for MPlayer'. There are buttons for 'Unstar' (with 88 stars) and 'Watch'. A dropdown menu is open over the 'master' branch, showing the latest commit by 'tb0hdan' 12 months ago. Below the branch dropdown are links for 'View code' and 'Jump to file'. A preview of the 'README.md' file content is shown, featuring the heading 'mplayer.py at a glance' and a block of Python code:

```
>>> p = mplayer.Player()
>>> p.loadfile('/path/to/file.mkv')
>>> p.time_pos = 40
>>> print p.length
```

A link at the bottom left says 'Abrir "https://github.com/baudm/mplayer.py/find/master" em uma nova aba'.

Integração do MPlayer com Python



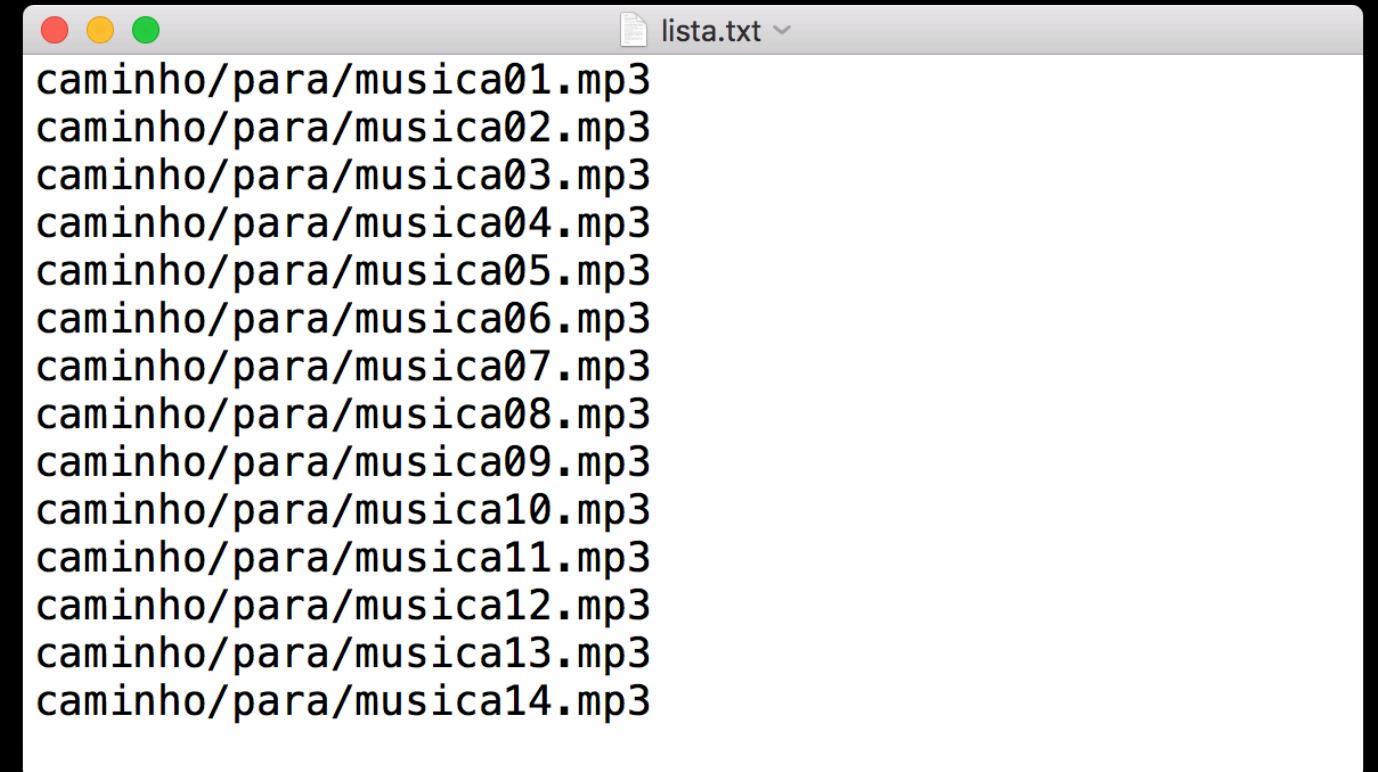
Foco da Aula: Reprodução de Áudio

```
>>> from mplayer import Player  
>>> player = Player()                                     ← inicializa o player  
>>> player.loadfile("Musica.mp3")  
>>> player.pause()  
>>> player.paused  
True  
>>> player.pause()                                     ← pause() serve para pausar e despausar  
>>> player.paused  
False  
>>> player.stop()  
>>> player.quit()                                     ← importante sempre sair do player no final!
```

```
>>> from mplayer import Player  
>>> player = Player()  
>>> player.loadfile("Musica.mp3")  
>>> player.length  
490.985944  
← duração total do áudio em segundos  
>>> player.time_pos  
145.358277  
← tempo atual do áudio em segundos  
>>> player.time_pos = 60  
← pula para um tempo específico  
>>> player.time_pos += 5  
← pula 5 segundos  
>>> player.quit()
```

```
>>> from mplayer import Player  
  
>>> player = Player()  
  
>>> player.loadfile("Musica.mp3")  
  
>>> player.metadata          ← retorna dicionário com metadados  
{'Title': 'Ixtapa', 'Artist': 'Rodrigo y Gabriela',  
'Album': 'Area 52'}  
  
>>> player.metadata['Artist']  
'Rodrigo y Gabriela'  
  
>>> player.metadata['Genre']    ← chave pode não existir nos  
Traceback (most recent call last):  
  File "<pyshell>", line 1, in <module>  
KeyError: 'Genre'  
  
>>> 'Genre' in player.metadata  
False  
  
>>> player.quit()
```

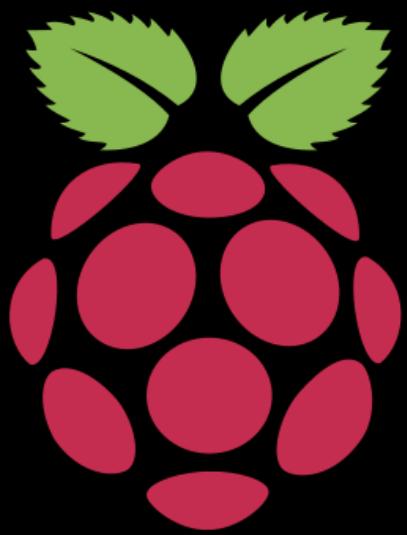
```
>>> from mplayer import Player  
>>> player = Player()  
  
>>> player.loadlist("lista.txt")  
  
>>> player.pt_step(1)           ← vai para a próxima música da lista  
  
>>> player.pt_step(2)  
  
>>> player.pt_step(-1)          ← vai para a música anterior da lista  
  
>>> player.quit()
```



```
>>> player.volume           ← valor entre 0 e 100  
100  
  
>>> player.volume = 70  
  
>>> player.speed = 2        ← toca 2 vezes mais rápido  
  
>>> player.speed = -2       ← toca 2 vezes mais lento
```

# Resumo da Ópera

| Funcionalidade     | Comandos  |
|--------------------|---|
| Entrada / Saída    | <code>x = input('Digite um número: ') • print('Resultado: ', x)</code>  |
| Listas             | <code>lista = [1, 2, 3] • lista2 = ['texto', [0, 0], 5]; lista[0] • lista.append(elemento) • lista.remove(indice)</code><br><u>mais funções na documentação oficial</u>   |
| Dicionários        | <code>dicionario = {'chave 1': 42, 'chave 2': [1, 2, 3], ...} dicionario['chave 1'] • dicionario['chave 3'] = 'Olá!'</code><br><u>mais funções na documentação oficial</u>  |
| Textos             | <code>x = 'texto' • y = "outro texto" • x + '\n' + y</code>   |
| Condicionais       | <code>if x == 0:<br/>    y = 4<br/>else:<br/>    if x not in [1, 2]:<br/>        y = 4<br/>    else:<br/>        y = 0</code><br><code>if x != 0<br/>    y = 4<br/>elif x &gt;= 0:<br/>    y = 3<br/>else:<br/>    y = 0</code> |
| Repetições         | <code>for i in [1, 2, 3] • for i in range(1, 4) • while x &gt; 1<br/>... • ...</code>   |
| Criação de Funções | <code>def funcao1(x):<br/>    return x + 2</code> <code>def funcao2(x, y, z):<br/>    ...</code> <code>def funcao3():<br/>    ...</code>  |



# Projeto 01

## Prática

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

ENG 1419 – Programação de Microcontroladores

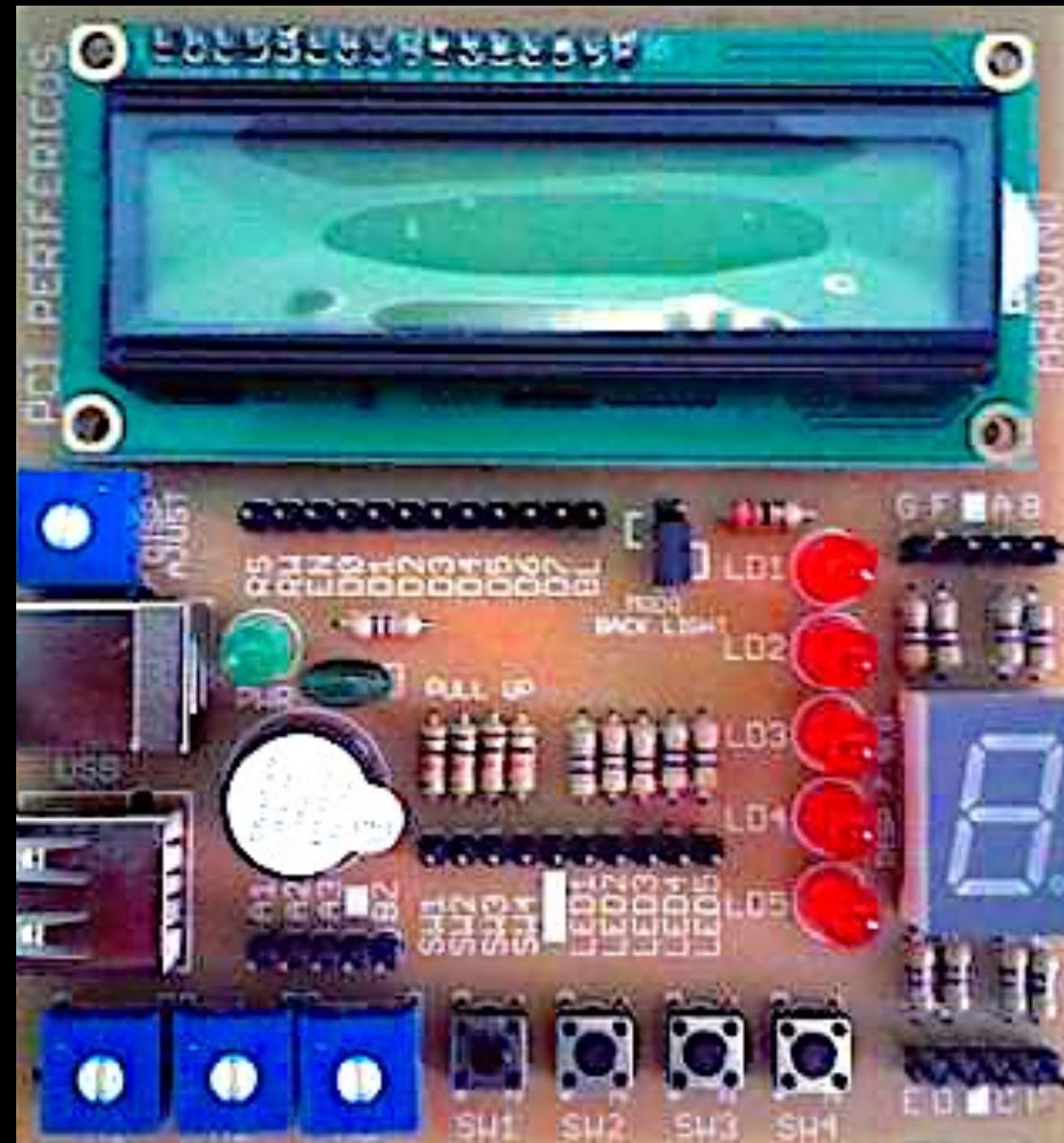


[janks.link/micro/projeto01.zip](https://janks.link/micro/projeto01.zip)

Material do Projeto 01

# Testes Iniciais

GPIO 2, 3, 4, 5, 6, 7



GPIO 21  
GPIO 22  
GPIO 23  
GPIO 24  
GPIO 25

GPIO 11, 12, 13, 14

Conexões dos Botões, LEDs e LCD de Caracters com as Portas da GPIO



## Testes Iniciais

Mude o estado do LED 1 (aceso/apagado) cada vez que alguém apertar o Botão 1

Pisque 4 vezes o LED 2 ao apertar o Botão 2.

Pisque continuamente o LED 3, mantendo-o aceso por meio segundo e apagado por 2 segundos.

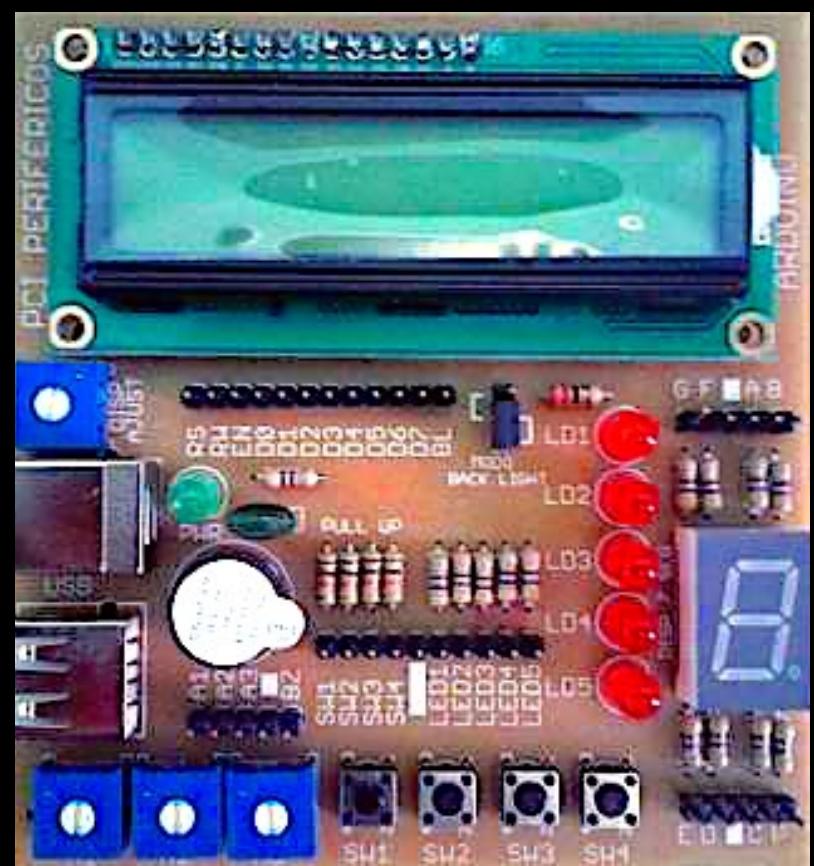
Mostre no LCD de caracteres quantas vezes o Botão 2 foi apertado.

↪ DICA: use uma variável global.

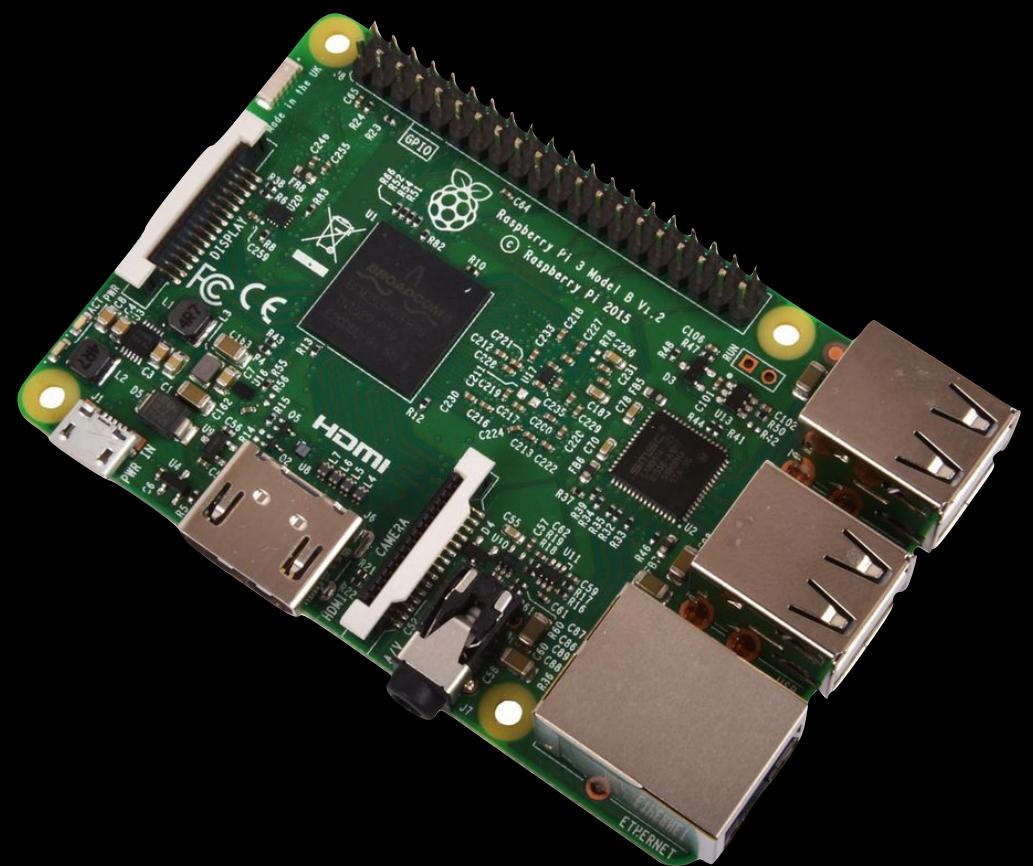
Acenda o LED 5 se e somente se os botões 3 e 4 estiverem simultaneamente pressionados

↪ DICA: use o while.

# Implementação



+

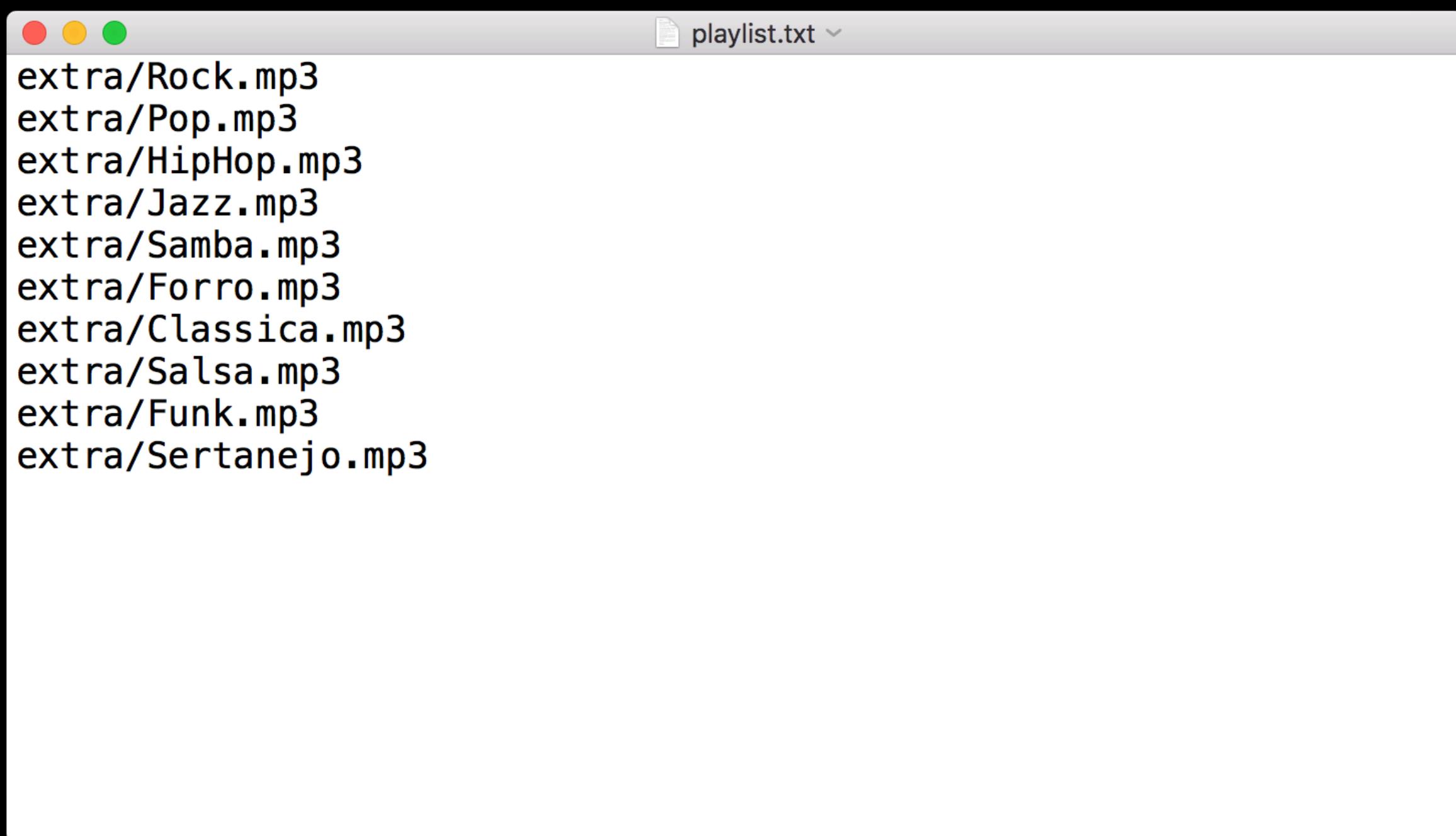


==



piPod

Controle de Mídia



Lista de Reprodução



## Implementação

Implemente a função de Tocar / Pausar. Use um LED para indicar se está tocando (mantendo aceso) ou pausado (piscando continuamente).

Implemente a função de avançar para Próxima Faixa.

Implemente a função de ir para a Faixa Anterior. Se a faixa atual não estiver no começo, o player deve voltar ao começo dela.

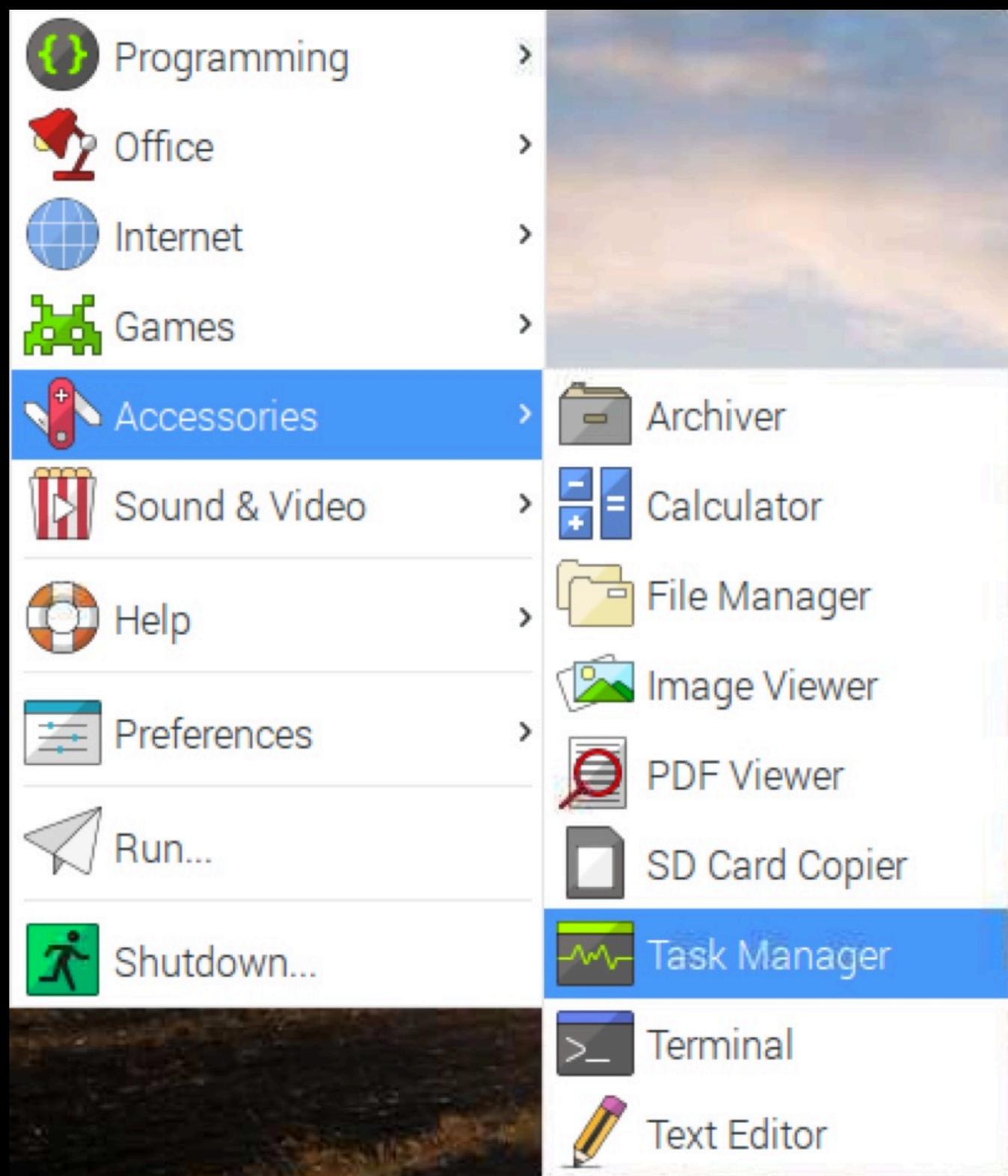
Exiba o nome da faixa atual na linha superior do LCD de caracteres.

↪ DICA: crie uma função e chame-a dentro de um `while` com `sleep`.

Exiba o tempo atual e o tempo total da música na linha inferior do LCD de caracteres (formato XX:XX)

↪ DICA: pesquise no Google como converter de segundos para o formato minutos:segundos.

caso o som pare de funcionar  
(ou haja mais de uma música ao mesmo tempo)



| Task Manager |      |      |         |       |                       |
|--------------|------|------|---------|-------|-----------------------|
| Command      | User | CPU% | RSS     | VM-Si |                       |
| python3      | pi   | 12%  | 16.2 MB | 73.6  |                       |
| thonny       | pi   | 1%   | 36.3 MB | 96.6  |                       |
| Ixtask       | pi   | 0%   | 17.5 MB | 35.9  |                       |
| mplayer      | pi   | 0%   | 26.2 MB | 86.4  | <span>Stop</span>     |
| Ixpanel      |      | 0%   | 25.5 MB | 93.2  | <span>Continue</span> |
| pcmanfm      |      | 0%   | 19.0 MB | 83.2  | <span>Term</span>     |
| vncserverui  |      | 0%   | 11.9 MB | 25.4  | <span>Kill</span>     |

Encerramento Forçado do MPlayer

# Aperfeiçoamento



## Aperfeiçoamento

Implemente um controle de 5 níveis de volume (de 0 ao máximo). Use 4 LEDs para indicar o nível atual.

Acelere a faixa ao segurar o botão de Avançar, e volte ao normal ao soltar.

↪ DICA: olhe a [documentação da gpiozero](#).

Role o nome da faixa no topo caso ela seja maior que o display.

↪ DICA: use a notação texto[inicio:fim] para capturar um trecho do texto.

código

TypeError: 'NoneType' object  
is not subscriptable

TypeError: '>' not supported between  
instances of 'NoneType' and 'int'

try:

código

```
except (AttributeError, TypeError) as error:  
    print('Deu ruim, mas bola para frente!')
```

Solução de um Possível Erro com o MPlayer