

Aplicações de SBCs: Caninos Loucos Labrador 32

Unidade 5 | Capítulo 2 | Objeto 1

Executores:



Coordenação:



Iniciativa:



Sumário



Introdução

GPIO

GPIO: 01-Blink

GPIO: 02-Toggle

Conclusão

Introdução

GPIOs Labrador 32:

Cada dispositivo é responsável por uma porta:

```
#/dev/gpiochip0 - Porta A
#/dev/gpiochip1 - Porta B
#/dev/gpiochip2 - Porta C
#/dev/gpiochip3 - Porta D
#/dev/gpiochip4 - Porta E
```

A imagem apresenta um texto com uma lista indicando a correspondência entre dispositivos e portas GPIO (General Purpose Input/Output). Segue a descrição detalhada:

Conteúdo do Texto:

Introdução:

O texto inicial diz: "Cada dispositivo é responsável por uma porta".

Lista de Dispositivos e Portas:

O texto associa os dispositivos aos nomes das portas, conforme mostrado abaixo:

/dev/gpiochip0 - Porta A

/dev/gpiochip1 - Porta B

/dev/gpiochip2 - Porta C

/dev/gpiochip3 - Porta D

/dev/gpiochip4 - Porta E

Formato Visual

O texto está escrito em fonte monoespaçada, típica de trechos de código ou configuração técnica.

As linhas começam com o símbolo #, indicando que são comentários em linguagens de programação ou documentação técnica.

Resumo:

A imagem descreve a associação entre diferentes dispositivos GPIO (gpiochip) e suas respectivas portas (de A a E).

Bibliotecas python-periphery:

- A biblioteca python-periphery oferece acesso direto e eficiente a periféricos de hardware em sistemas Linux, como GPIO, I2C, SPI, UART, PWM e MMIO, sendo ideal para projetos IoT e automação. Simples e de alto desempenho, facilita o controle de sensores e atuadores em dispositivos embarcados.

Introdução



Pesquisar projetos

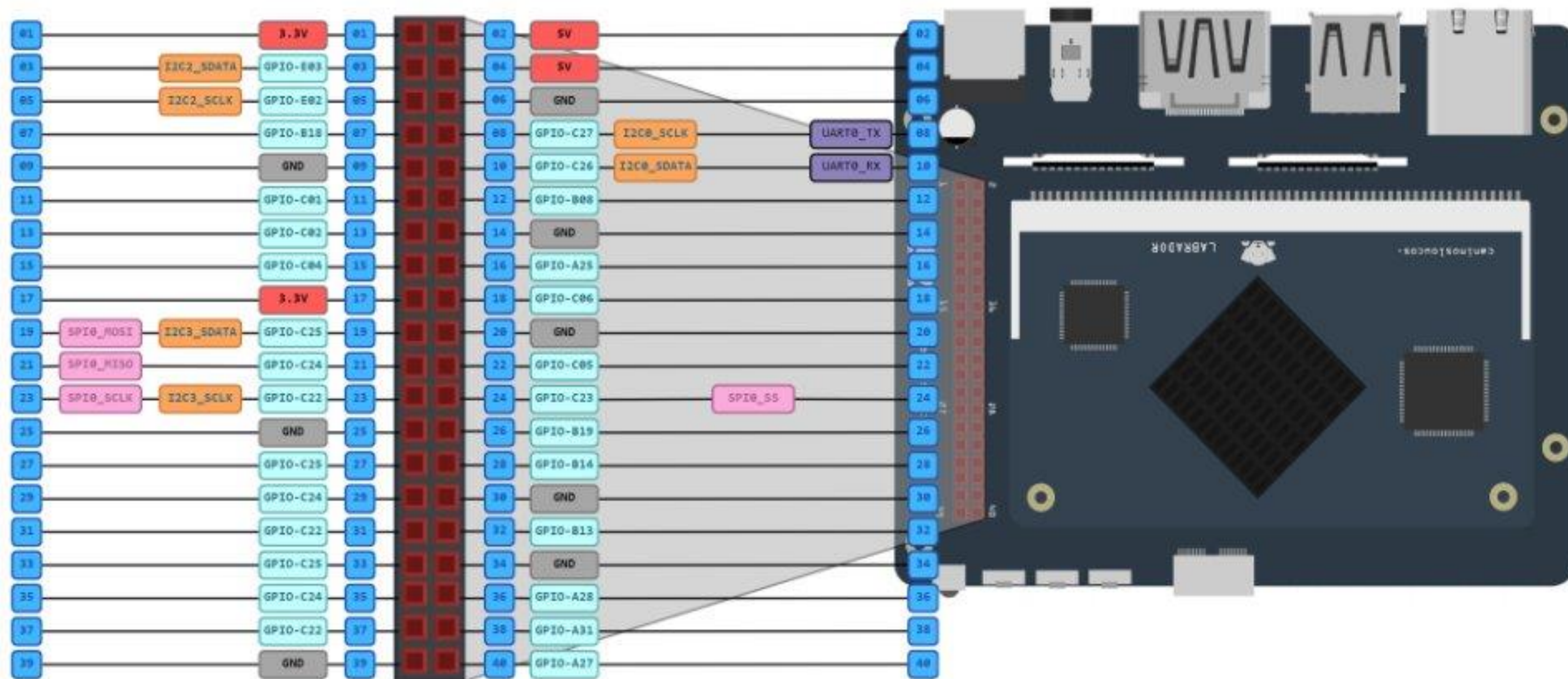
python-periphery 2.4.1

```
pip install python-periphery
```



Labrador 32 - GPIO

» https://www.canva.com/design/DAGcMFy6Sr0/rm6CzbnHj7wLVhNnz0sguQ/edit?utm_content=DAGcMFy6Sr0&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton



Labrador 32 - GPIO

- Acesso aos Dispositivos de GPIO /dev/gpiochip:

Antes de executar seu código, execute os seguintes

comandos: `sudo chown caninos /dev/gpiochip*`

`sudo chmod g+rw /dev/gpiochip*`

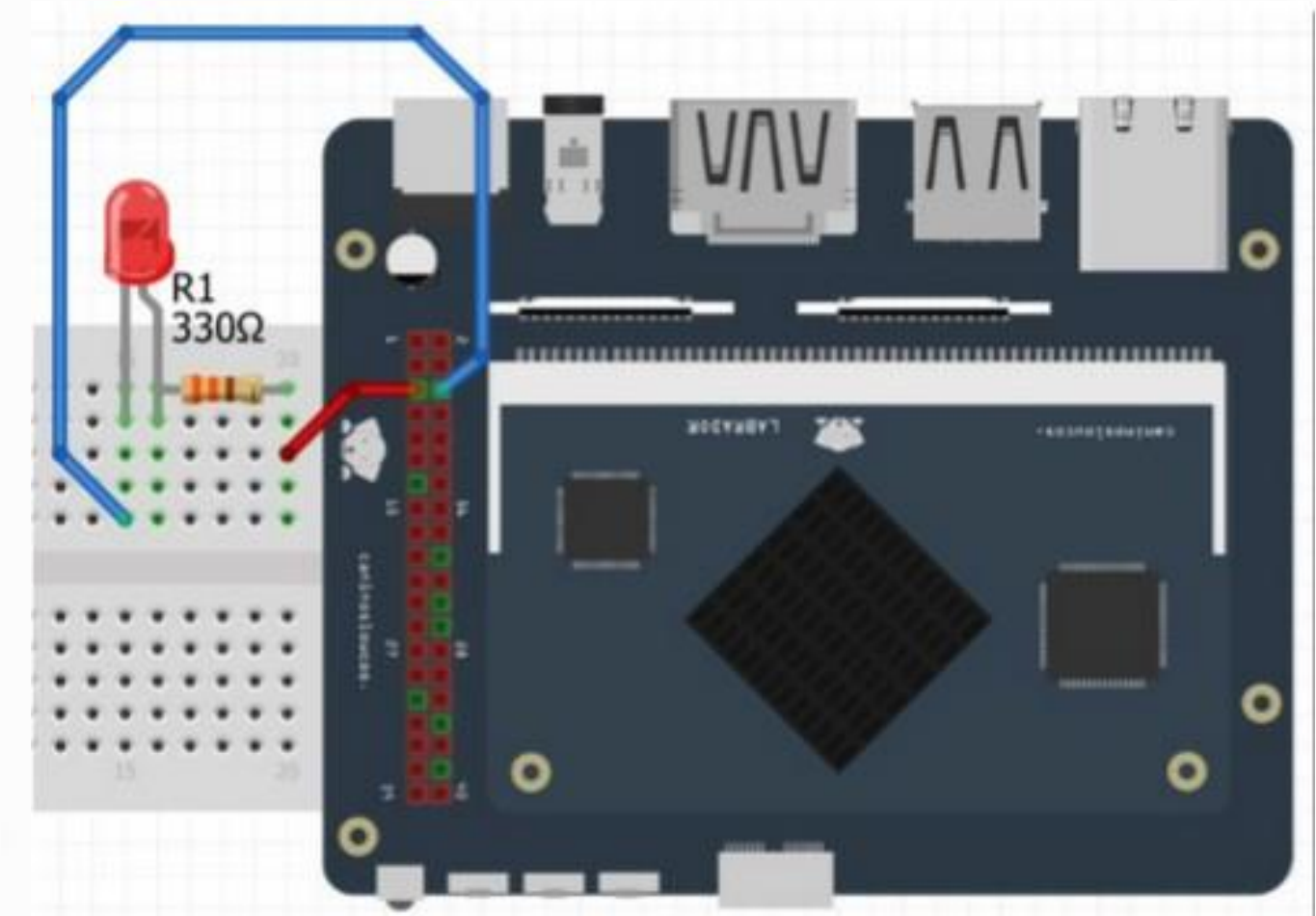
Esses comandos adicionam e configuram privilégios temporários para acessar o dispositivo de gpio.

Labrador 32 - GPIO: 01-Blink

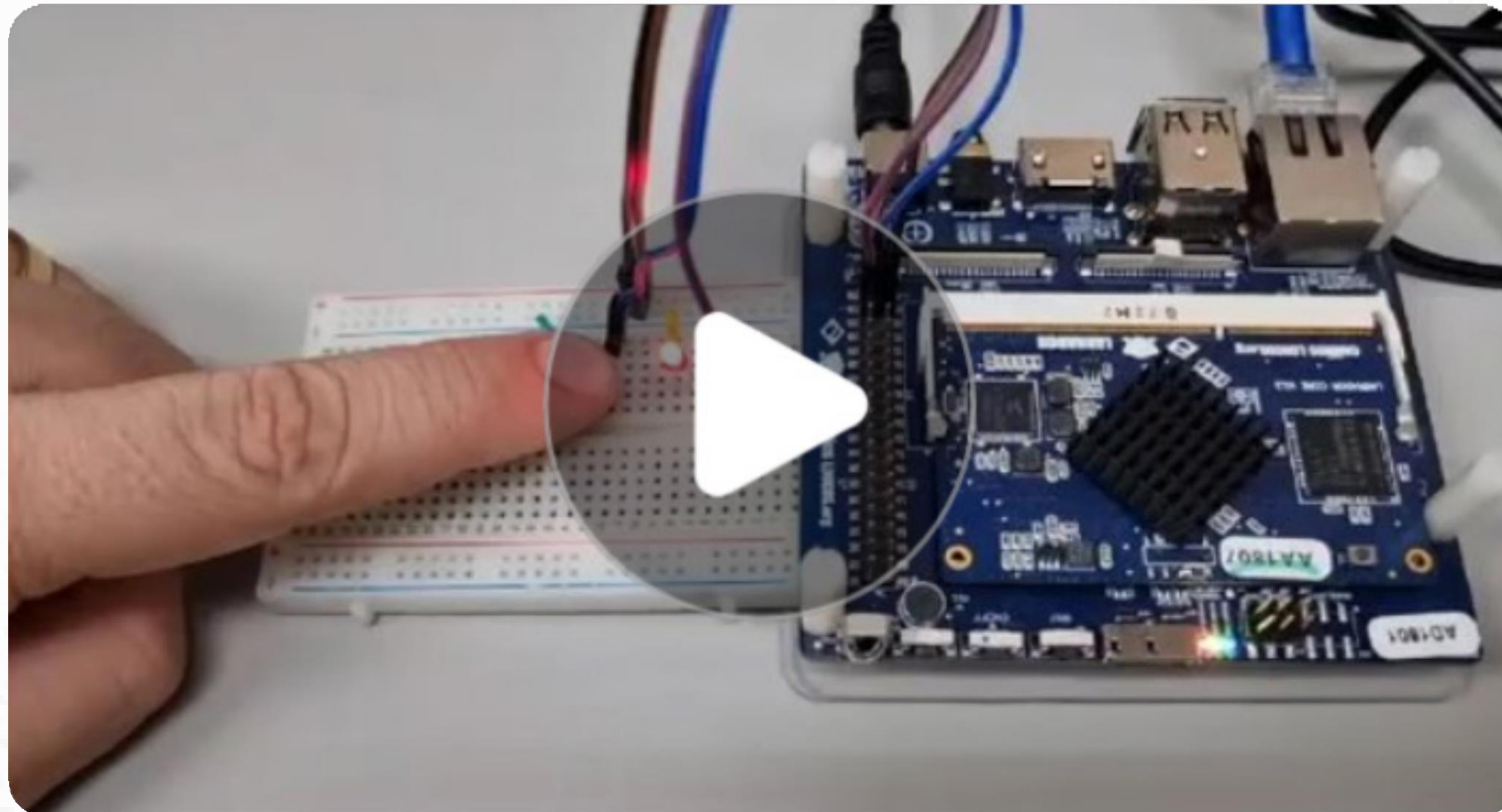
- Configuração de um pino como saída para controlar um LED:

```
from periphery import GPIO  
import time  
  
# Configura a variável LED - como na GPIO /dev/  
gpiochip4 (E) linha 2 como saída - GPIO-E2  
(Header-5)  
led = GPIO("/dev/gpiochip4", 2, "out") #  
while True:  
    led.write(True)  
    time.sleep(1)  
    led.write(False)  
    time.sleep(1)
```

```
#Pino 06: GND  
#Pino 05: GPIO Output  
  
#R1: Resistor limitador de corrente
```



Labrador 32 - GPIO: 01-Blink



Labrador 32 - GPIO

- Para evitar habilitar os privilégios toda vez que iniciar a labrador, iremos criar uma Regra Udev:
 - » **s sudo nano /etc/udev/rules.d/99-gpio.rules**
 - » **SUBSYSTEM=="gpio", GROUP="caninos", MODE="0660"**
 - » **KERNEL=="gpiochip*", GROUP="caninos", MODE="0660" Salvar e sair:**

Pressione CTRL +O, Enter para salvar, e CTRL +X para sair.

- » **s sudo udevadm control --reload-rules**
- » **s sudo udevadm trigger**

Labrador 32 - GPIO

Adicionar usuário ao grupo:

- » **s sudo usermod -aG caninos caninos**
- » **s sudo reboot**

Verificar as Alterações:

- » **s ls -l /dev/gpiochip***

Labrador 32 - GPIO: 02-Toggle

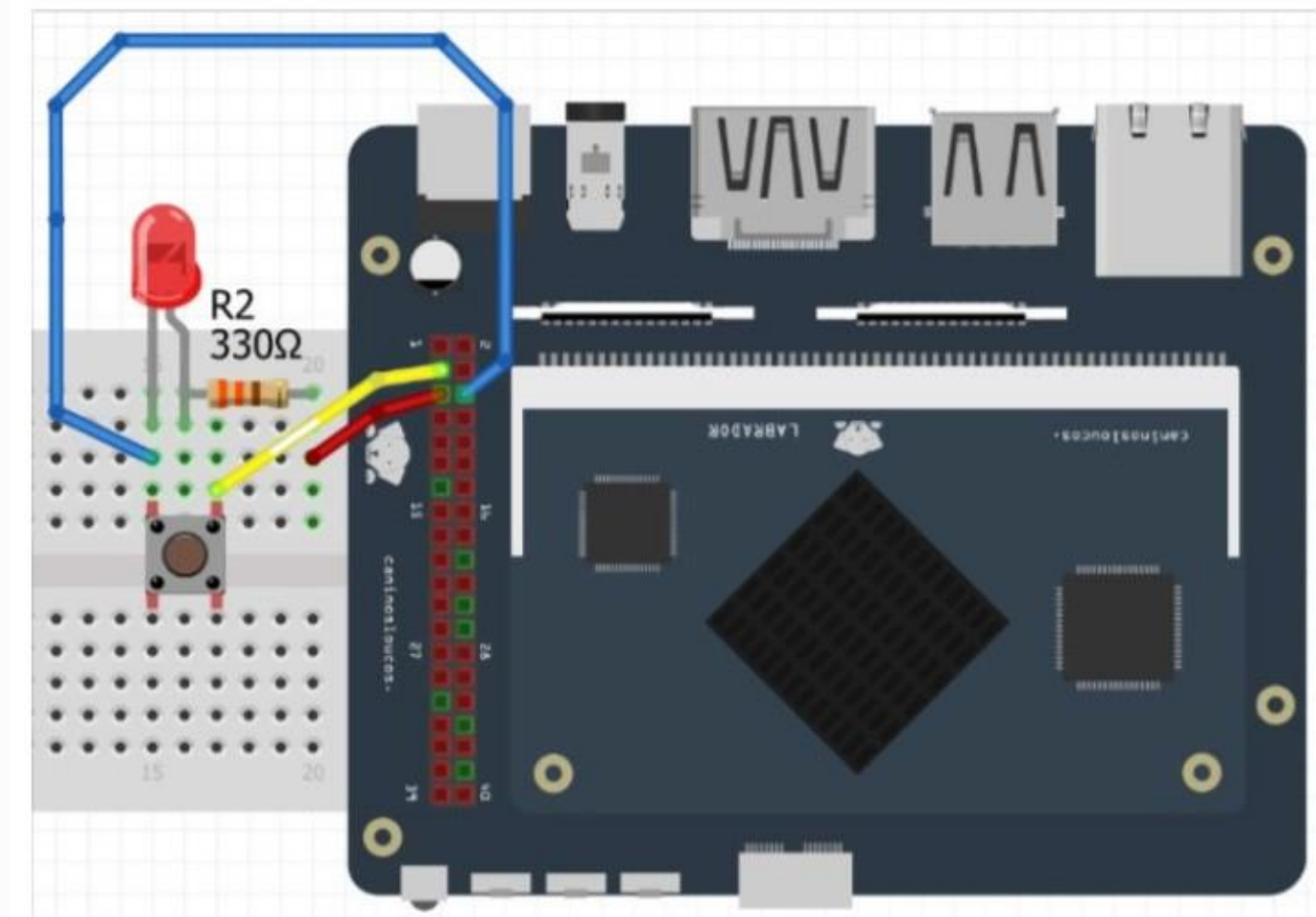
- Configuração de um pino como entrada para ler o estado de um botão e alterar o estado do LED

```
from periphery import GPIO
import time

# Configura o pino LED - GPIO /dev/gpiochip4; linha 2 (E2) como
saída - GPIO-E2 (Header-5)
led = GPIO("/dev/gpiochip4", 2, "out")

# Configura o pino Button - GPIO /dev/gpiochip4 line 3 (E3) como
entrada - GPIO-E3 (Header-3)
button = GPIO("/dev/gpiochip4", 3, "in", bias="pull_up")

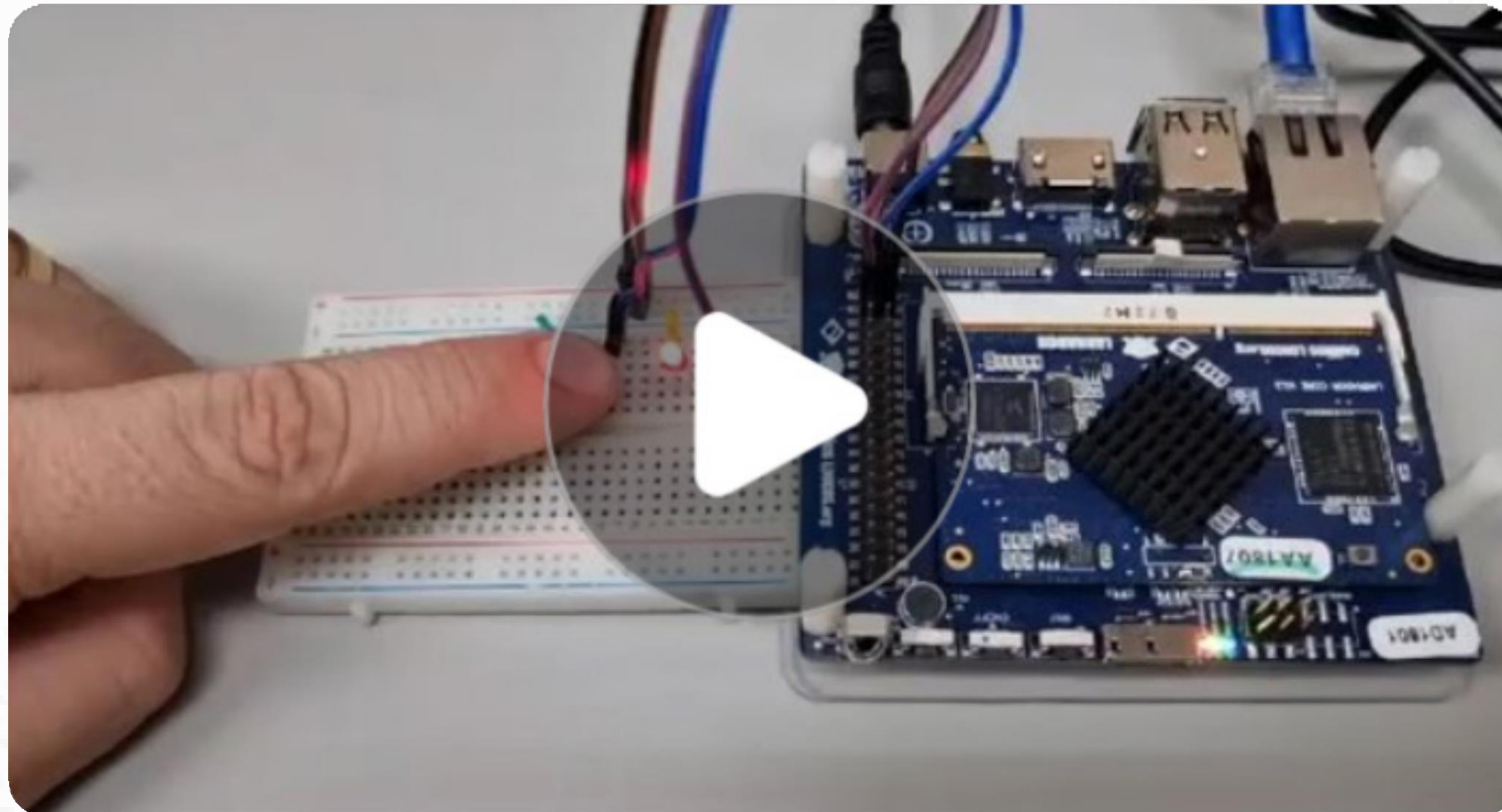
try:
    while(1):
        if(not button.read()): # Botão pressionado:
            led.write(not led.read())
            time.sleep(0.3) # Filtragem de bounce
finally:
    button.close()
    led.close()
```



```
#Pino 06: GND
#Pino 03: 3V3
#Pino 05: Saída

#R1: Resistor limitador
#Pull-Up: Interno
```

Labrador 32 - GPIO: 01-Blink



Recaptação

Resumo dos Tópicos Abordados:

- As capacidades da Labrador no controle de GPIOs;
- Conceito dos dispositivos gpiochip;
- Característica de portas de pinos de entradas e saídas;
- Exemplos práticos utilizando a biblioteca python-periphery.
 - » <https://github.com/jorgewattes/labrador-examples/tree/main/embarcatech>

Próximos Passos:

- Comunicações seriais UART, I2C, SPI e suas possibilidades.

Aplicações de SBCs: Caninos Loucos Labrador 32