



2023

4. Controlador de semáforo

Número do projeto: **2021-1-FR01-KA220-SCH-000031617**



 Co-funded by
the European Union

O apoio da Comissão Europeia à produção desta publicação não constitui um endosso do conteúdo, que reflete apenas as opiniões dos autores, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito das informações nele contidas.

Parceria SCRAPY
31/05/2023



Índice

Experiência 4: Controlador de semáforos	2
Objetivos	2
Materiais a utilizar	2
Passos a seguir	3
Diagrama de ligação	4
Código	5
Conclusão	6

Experiência 4: Controlador de semáforos

Breve Descrição

Com esta experiência, os alunos serão capazes de controlar luzes LED que se acendem na mesma ordem de um semáforo.

Descrição Estendida

Nesta atividade utilizaremos 3 LED em cores diferentes (vermelho, amarelo e verde) e os alunos utilizarão um botão para reiniciar o circuito.

O Raspberry Pi Pico deteta a mudança de nível do botão para determinar se o botão foi pressionado. Pressione o botão para ligar a luz LED pela primeira vez e pressione o botão para fazer com que o LED vermelho volte a acender, de modo a perceber a função de ligar e desligar a luz LED, bem como cronometrar as luzes para que elas acendam uma após a outra.

Objetivos

Com esta atividade, os alunos irão experimentar um botão de pressão, LED de diferentes cores e temporização para que o semáforo funcione sem problemas.

Em termos de conhecimentos, os alunos irão:

1. Entenda o que é um circuito.
2. Ser capaz de identificar o hardware usado em um circuito.
3. Crie três LEDs (vermelho, amarelo e verde) e aprenda a codificá-los para acender sequencialmente.
4. Adicione um botão ao sistema e entenda como ele pode funcionar com ele.

Materiais a utilizar

- 1 x Raspberry Pi Pico
- 1 x kit de breadboard Pico
- 1 x Breadboard de tamanho normal
- 1 x OLED I2C ICC
- 1 x botão
- 3 x LED (vermelho, verde e amarelo)
- 1 x buzzer
- 3 x resistor de 220 ohm
- Fios de jumper

Passos a seguir

Vamos criar um semáforo que se move sequencialmente e usar um botão para reiniciar o circuito.

Para isso, vamos começar por

1. Conecte o **OLED I2C ICC** à placa Raspberry Pi Pico usando fios de conexão.
2. Conecte o botão à placa Raspberry Pi Pico.
3. Conecte o buzzer e os LEDs à placa Raspberry Pi Pico usando fios de conexão e as resistências de 220 ohm para limitar o fluxo de corrente.
4. Escreva um programa Python para controlar a placa Raspberry Pi Pico e use o botão para iniciar/reiniciar o semáforo.
5. Teste o botão para ver o que acontece com os LEDs e o OLED I2C ICC.

Placa Raspberry Pi Pico:

- GP26: Pino SDA do OLED I2C ICC
- GP27: Pino SCL do OLED I2C ICC
- GP7: Pino 1 do botão
- GP39: Pino 3 do botão
- GP13: Pino positivo do LED Vermelho
- GP12: Pino positivo do LED amarelo
- GP11: Pino positivo do LED verde
- GP16: Pino positivo da campainha
- GND: Pino de terra da placa

OLED I2C ICC:

- VCC: Conecte a 3V3/5V da placa Raspberry Pi Pico
- GND: Conecte ao GND da placa Raspberry Pi Pico
- SCL: Conecte ao GP27 da placa Raspberry Pi Pico
- SDA: Conecte ao GP26 da placa Raspberry Pi Pico

Botão de pressão:

- Pino 1: Conecte ao GP7 da placa Raspberry Pi Pico através de uma resistência de 220 ohm
- Pino 3: Conecte ao 3V3 da placa Raspberry Pi Pico

LED vermelho:

- Perna positiva: Conecte ao GP13 da placa Raspberry Pi Pico através de uma resistência de 220 ohm
- Perna negativa: Conecte ao GND da placa Raspberry Pi Pico

LED amarelo:

- Perna positiva: Conecte ao GP12 da placa Raspberry Pi Pico através de uma resistência de 220 ohm
- Perna negativa: Conecte ao GND da placa Raspberry Pi Pico

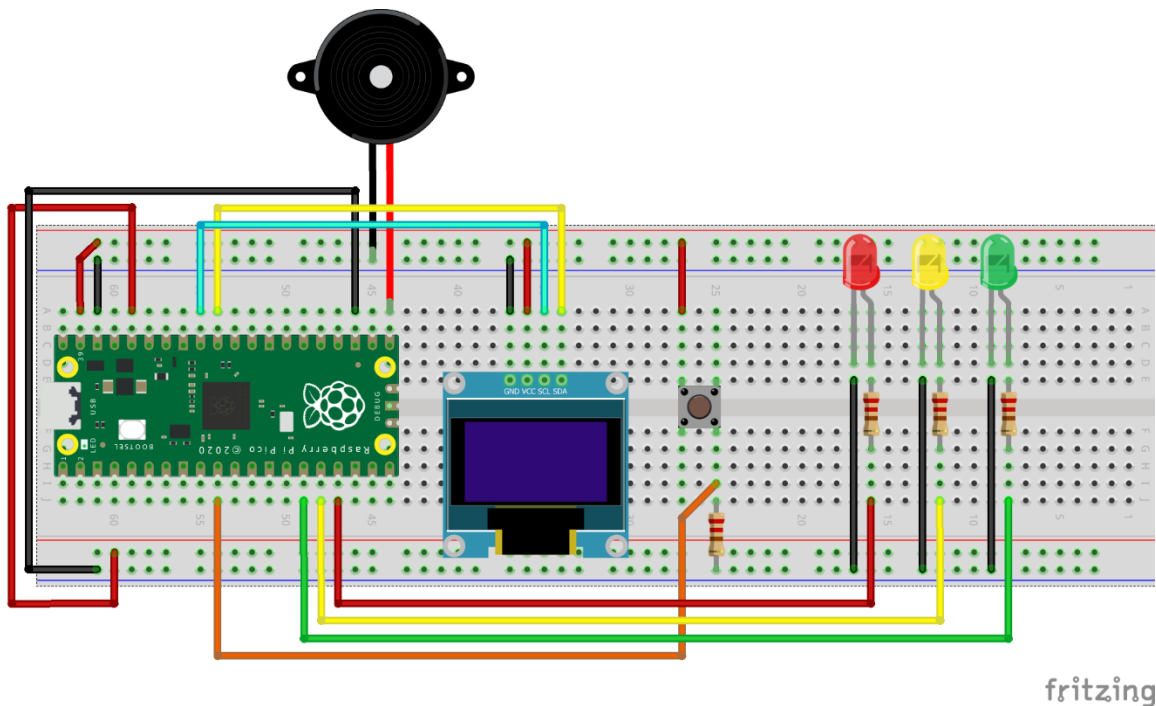
LED verde:

- Perna positiva: Conecte ao GP11 da placa Raspberry Pi Pico através de uma resistência de 220 ohm
- Perna negativa: Conecte ao GND da placa Raspberry Pi Pico

Buzzer:

- Perna positiva: Conecte ao GP16 da placa Raspberry Pi Pico
- Perna negativa: Conecte ao GND da placa Raspberry Pi Pico

Diagrama de ligação



fritzing

Código

```
import machine
import ssd1306
import utime

# Atribuições de pinos
button_pin = machine.Pin(7, machine.Pin.IN,
machine.Pin.PULL_DOWN)
red_led_pin = machine.Pin(13, machine.Pin.OUT)
yellow_led_pin = machine.Pin(12, machine.Pin.OUT)
green_led_pin = machine.Pin(11, machine.Pin.OUT)
buzzer_pin = machine.Pin(16, machine.Pin.OUT)

# Inicializar exibição OLED
i2c = machine.I2C(0, sda=machine.Pin(0), scl=machine.Pin(1))
oled = ssd1306.SSD1306_I2C(128, 32, i2c)

# Definir estado inicial
is_crossing_allowed = False

def button_interrupt_handler(pin):
    global is_crossing_allowed
    if pin.value() == 1:
        is_crossing_allowed = not is_crossing_allowed

# Interrupção do botão Registrar
button_pin.irq(trigger=machine.Pin.IRQ_RISING,
handler=button_interrupt_handler)

# Função para atualizar o visor
def update_display():
    oled.text("TRAFFIC LIGHT", 0, 0)
    if is_crossing_allowed:
        oled.text("CROSSING:", 0, 12)
        oled.text("ALLOWED", 0, 22)
    else:
        oled.text("PLEASE", 0, 12)
        oled.text("WAIT", 0, 22)
    oled.show()
```

```
# Função para controlar o semáforo
def control_traffic_light():
    red_led_pin.value(1)
    yellow_led_pin.value(0)
    green_led_pin.value(0)
    buzzer_pin.value(1)
    utime.sleep(4) #control time
    buzzer_pin.value(0)
    red_led_pin.value(0)
    yellow_led_pin.value(0)
    green_led_pin.value(1)
    utime.sleep(4) #control time
    red_led_pin.value(0)
    yellow_led_pin.value(1)
    green_led_pin.value(0)
    utime.sleep(0.5)

# Loop principal
while True:
    control_traffic_light()
    update_display()
```

Conclusão

Se necessário, o professor pode desafiar os alunos a criar o código para que o botão controle a cor do LED que acende e usar este programa como uma ferramenta de avaliação para a aplicação do quadro criado.

Desta forma, o professor apresenta uma pergunta, e os alunos respondem acendendo um adesivo vermelho, amarelo ou verde de acordo com o que eles acham que é a resposta. Com este recurso, o professor pode ver imediatamente os resultados da escolha do aluno.