



2023

1. Sensor de backup de carro DIY

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 1: Sensor de backup DIY Car | 2 |
| Objetivos | 3 |
| Materiais a utilizar | 3 |
| Passos a seguir | 4 |
| Diagrama de ligação | 4 |
| Código | 6 |
| Exemplos de imagens | 7 |
| Conclusão | 8 |

Experiência 1: Sensor de backup DIY Car

Breve Descrição

Crie um sistema de inversão de marcha-atrás com Raspberry Pi Pico e sensor ultrassônico.

Descrição Estendida

Você acha um pouco complicado inverter seu carro? Você se preocupa em acertar algo ou alguém durante o backup? Bem, não se preocupe mais! Com a ajuda da placa Raspberry Pi Pico e de um sensor ultrassônico HC-SR04, pode construir o seu próprio sistema de radar de marcha-atrás que tornará o estacionamento do seu carro muito fácil.

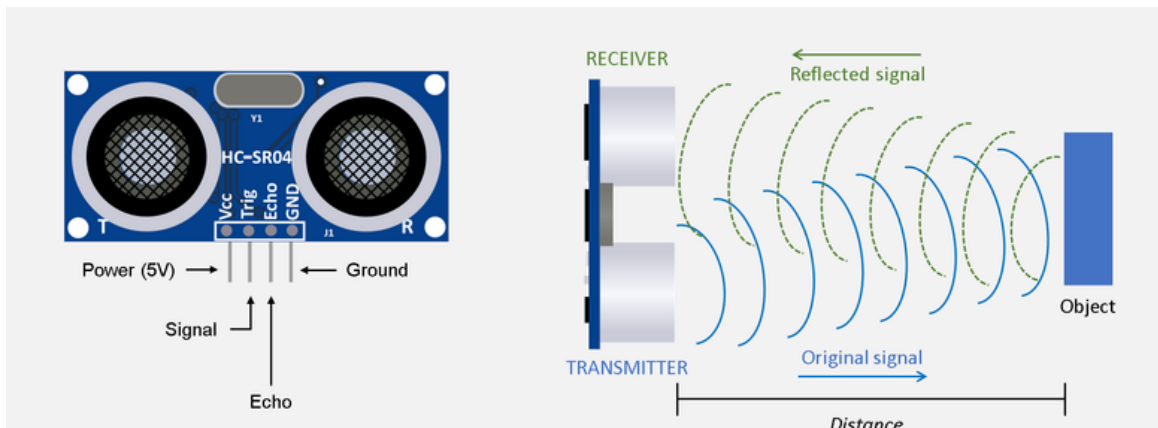
- Este projeto DIY é perfeito para qualquer pessoa interessada em eletrônica e engenharia, e é adequado para todos os níveis de habilidade. Este radar reversível irá certamente desafiá-lo e encantá-lo.
- Além do sensor Raspberry Pi Pico e HC-SR04, este projeto também requer um buzzer, três LEDs (verde, amarelo e vermelho), três resistores de 220 ohm, uma placa de prototipagem rápida e fios de conexão. Com estes componentes, poderá criar um radar de marcha-atrás fiável e preciso que o ajudará a estacionar o seu carro com facilidade.
- Seguindo as instruções passo a passo, você aprenderá como conectar os componentes, programar a placa Raspberry Pi Pico e testar o sistema para garantir que ele funcione corretamente. Você também aprenderá sobre a física dos sensores ultrassônicos e como eles podem ser usados para medir a distância.

Os princípios operacionais do sensor ultrassônico HC-SR04?

O sensor ultrassônico HC-SR04 funciona emitindo ondas sonoras de alta frequência que são inaudíveis para os ouvidos humanos. Essas ondas sonoras viajam pelo ar e saltam de objetos em seu caminho. Quando as ondas sonoras atingem um objeto, elas são refletidas de volta para o sensor, que mede o tempo que leva para as ondas se recuperarem. Ao saber a velocidade do som e o tempo que leva para as ondas viajarem até o objeto e vice-versa, o sensor pode calcular a distância até o objeto.

Aqui estão os principais passos nos princípios operacionais do sensor ultrassônico HC-SR04:

- sensor envia uma onda sonora de alta frequência (normalmente cerca de 40 kHz) do seu transmissor.
- A onda sonora viaja pelo ar e salta de um objeto.
- A onda sonora refletida é detetada pelo recetor do sensor.
- O sensor mede o tempo que a onda sonora demora a viajar até ao objeto e vice-versa.
- O sensor calcula a distância ao objeto com base no tempo que a onda sonora levou para viajar até o objeto e vice-versa.



No geral, o sensor ultrassônico HC-SR04 é uma maneira confiável e precisa de medir distâncias, e é comumente usado em aplicações como robótica, automação e detecção de distância.

Objetivos

Através desta atividade, o usuário experimentará a construção de um sistema de radar reversível usando a placa Raspberry Pi Pico e um sensor ultrassônico HC-SR04. O utilizador irá adquirir conhecimentos sobre:

- A física das ondas ultrassônicas e como elas podem ser usadas para medir a distância.
- O básico de programação em Python e como escrever código para controlar a placa Raspberry Pi Pico.
- Os princípios do projeto de circuitos e como conectar componentes juntos em uma placa de prototipagem rápida para criar um sistema de radar de inversão funcional.

Ao concluir este projeto, o usuário obterá uma compreensão mais profunda de eletrônica, engenharia e programação. Eles também terão um dispositivo prático e útil que eles podem usar para tornar o estacionamento do seu carro mais seguro e mais conveniente.

Materiais a utilizar

- 1 x Raspberry Pi Pico
- 1 x kit de breadboard Pico
- 1 x Breadboard de tamanho completo
- 1 x sensor ultrassônico HC-SR04
- 1 x campainha
- 3 x LEDs (verde, amarelo e vermelho)
- 3 x resistências de 220 ohm
- Fios de jumper

Passos a seguir

Os principais passos para realizar o experimento de radar reversível com a placa Raspberry Pi Pico e sensor ultrassônico HC-SR04:

1. Conecte o sensor ultrassônico HC-SR04 à placa Raspberry Pi Pico usando fios de conexão.
2. 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.
3. Escreva um programa Python para controlar a placa Raspberry Pi Pico e use o sensor HC-SR04 para medir distâncias.
4. Programe a placa Raspberry Pi Pico para ligar o LED verde quando não houver obstáculo, o LED amarelo quando o obstáculo estiver dentro de um determinado alcance e o LED vermelho quando o obstáculo estiver muito perto.
5. Programe a placa Raspberry Pi Pico para ativar a campainha quando o obstáculo estiver muito perto.
6. Teste o sistema de radar de marcha-atrás colocando obstáculos a várias distâncias e ângulos atrás do radar e certifique-se de que os LEDs e o buzzer fornecem o feedback adequado.

Para o Passo 4: Para estudar o funcionamento do sistema, é necessário programar o seu comportamento. Os valores da distância podem ser ajustados de acordo com requisitos específicos. A programação deve seguir as seguintes regras:

- LED verde deve estar continuamente aceso se a distância entre o sensor e qualquer obstáculo for superior a 20cm (1 metro).
- O LED laranja deve estar aceso se a distância entre o sensor e qualquer obstáculo estiver entre 20cm e 5cm.
- O LED vermelho deve estar aceso se a distância entre o sensor e qualquer obstáculo for inferior a 5cm.

Ao programar o sistema para seguir essas regras, ele fornecerá feedback visual claro da distância entre o sensor e qualquer obstáculo.

Essas regras podem ser ajustadas conforme necessário para atender a requisitos específicos ou casos de uso.

Diagrama de ligação

Aqui está um esquema de fiação para o projeto de radar de inversão com placa Raspberry Pi Pico, sensor ultrassônico HC-SR04, buzzer e três LEDs:

Placa Raspberry Pi Pico:

- GP15: Pino de gatilho do sensor HC-SR04
- GP14: Pino de eco do sensor HC-SR04
- GP10: Pino positivo do LED verde

- GP11: Pino positivo do LED laranja
- GP12: Pino positivo do LED Vermelho
- GP2: Pino positivo da campainha
- GND: Pino de terra da placa

HC-SR04 Sensor:

- VCC: Conecte à fonte de alimentação de 5V
- GND: Conecte ao GND da placa Raspberry Pi Pico
- Trig: Conecte ao GP15 da placa Raspberry Pi Pico
- Echo: Conecte ao GP14 da placa Raspberry Pi Pico

LED verde:

- Perna positiva: Conecte ao GP10 da placa Raspberry Pi Pico através de uma resistência de 220 ohm
- Perna negativa: Conecte ao GND da placa Raspberry Pi Pico

LED laranja:

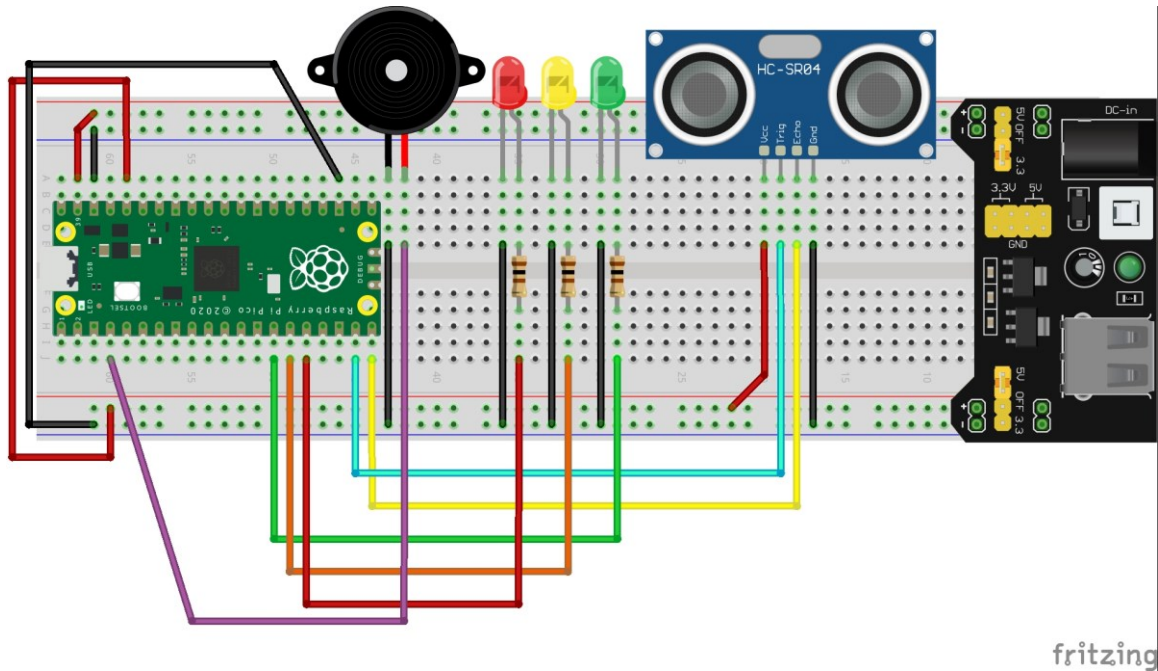
- Perna positiva: Conecte ao GP11 da placa Raspberry Pi Pico através de uma resistência de 220 ohm
- Perna negativa: Conecte ao GND da placa Raspberry Pi Pico

LED vermelho:

- 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

Buzzer:

- Perna positiva: Ligue ao GP2 da placa Raspberry Pi Pico
- Perna negativa: Conecte ao GND da placa Raspberry Pi Pico



Código

```
# Importar bibliotecas necessárias
from machine import Pin, time_pulse_us
import time

# Definir pinos para os componentes
trigger_pin = Pin(15, Pin.OUT)
echo_pin = Pin(14, Pin.IN)
green_led = Pin(10, Pin.OUT)
orange_led = Pin(11, Pin.OUT)
red_led = Pin(12, Pin.OUT)
buzzer = Pin(2, Pin.OUT)

# Definir os limites de distância para cada LED
green_threshold = 20 # cm
orange_threshold = 5 # cm

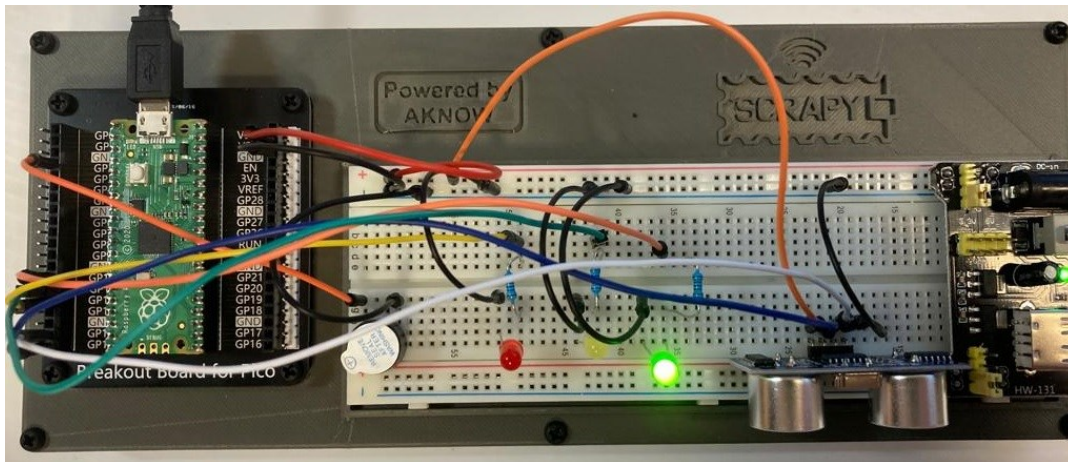
# Definir a função para calcular a distância do sensor HC-SR04
def get_distance():
    # Send a 10us pulse to trigger the sensor
    trigger_pin.low()
    time.sleep_us(2)
    trigger_pin.high()
    time.sleep_us(10)
    trigger_pin.low()
    # Meça a duração do sinal de eco
    duration = time_pulse_us(echo_pin, 1, 10000)
    # Calcular a distância da duração utilizando a velocidade do som (343 m/s)
    distance = duration / 2 / 1000000 * 343 * 100
    return distance

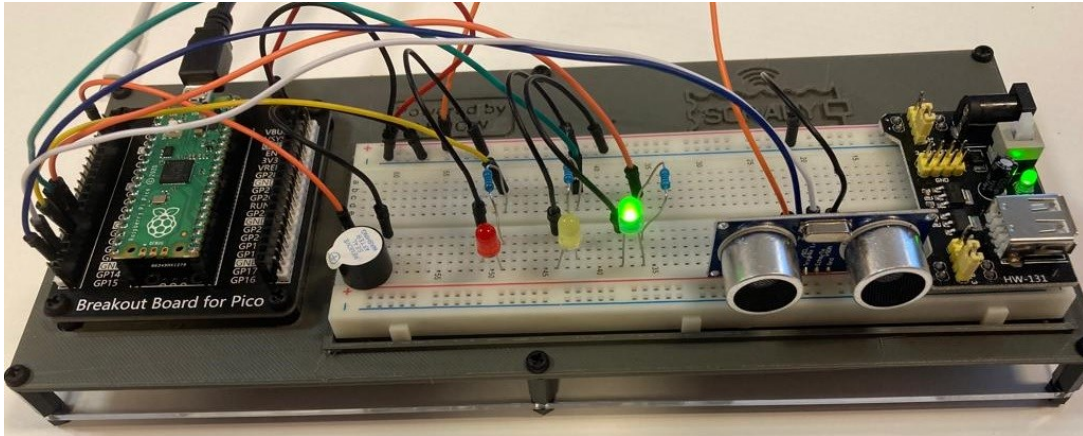
# Defina o loop principal para ler a distância e controlar os LEDs e buzzer
```



```
while True:
    # Obtenha a distância do sensor e imprima-o
    distance = get_distance()
    print(f"Distance : {distance} cm")
    # Ligue o LED verde se a distância for maior do que o limite
    if distance > green_threshold:
        green_led.on()
        orange_led.off()
        red_led.off()
        buzzer.off()
    # Ligue o LED laranja se a distância estiver entre os limites
    elif distance > orange_threshold:
        green_led.off()
        orange_led.on()
        red_led.off()
        buzzer.off()
    # Acenda o LED vermelho e a campainha se a distância for inferior ao limite
    else:
        green_led.off()
        orange_led.off()
        red_led.on()
        buzzer.on()
        time.sleep(0.5) # Buzz for 0.5 seconds
    # Aguarde 0,1 segundo antes da próxima medição
    time.sleep(0.1)
```

Exemplos de imagens





Conclusão

Em conclusão, este projeto envolveu o uso de uma placa Raspberry Pi Pico, um sensor ultrassônico HC-SR04 e alguns componentes adicionais para criar um radar de ré para carros. Aprendemos sobre os princípios de funcionamento do sensor HC-SR04, como ligar os componentes e como programar o comportamento do sistema usando MicroPython. Seguindo os passos deste tutorial, fomos capazes de criar um sistema que pode detetar obstáculos e fornecer feedback visual e sonoro ao motorista.

Quanto a uma maior exploração, há muitas maneiras de expandir este projeto. Algumas sugestões incluem:

- Adicionar mais LEDs ou um ecrã para fornecer feedback de distância mais detalhado.
- Usando algoritmos de aprendizado de máquina para melhorar a deteção de obstáculos e precisão.
- Criação de uma interface sem fio para permitir que o sistema se comunique com um dispositivo móvel ou outro dispositivo externo.
- Incorporação de sensores ou componentes adicionais para criar um sistema de segurança automóvel mais abrangente.

No geral, este projeto fornece uma ótima introdução ao mundo da eletrônica e da programação e serve como ponto de partida para mais exploração e experimentação.