



2023

1. DIY Capteur de recul de voiture

Projet Numéro: **2021-1-FR01-KA220-SCH-000031617**



**Co-funded by
the European Union**

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

SCRAPY Partnership

31/05/2023

Table des matières

Expérience 1 : Capteur de recul pour voiture	2
Description courte.....	2
Description détaillée	2
Objectifs :	3
Matériel à utiliser :	3
Étapes à suivre :	4
Schéma de câblage	4
Code.....	6
Photos	7
Conclusion.....	8

Expérience 1 : Capteur de recul pour voiture

Description courte

Créer un système de recul de voiture avec Raspberry Pi Pico et un capteur à ultrasons.

Description détaillée

Trouvez-vous que faire marche arrière avec votre voiture est un peu difficile ? Vous craignez de heurter quelque chose ou quelqu'un en reculant ? Eh bien, ne vous inquiétez plus ! Avec l'aide de la carte Raspberry Pi Pico et d'un capteur ultrasonique HC-SR04, vous pouvez construire votre propre système de radar de recul qui fera du stationnement de votre voiture un jeu d'enfant.

- Ce projet de bricolage est parfait pour tous ceux qui s'intéressent à l'électronique et à l'ingénierie, et il convient à tous les niveaux de compétence. Ce radar de recul ne manquera pas de vous mettre au défi et de vous ravir.
- Outre le Raspberry Pi Pico et le capteur HC-SR04, ce projet nécessite également un buzzer, trois LED (verte, jaune et rouge), trois résistances de 220 ohms, une carte de prototypage rapide et des fils de connexion. Grâce à ces composants, vous serez en mesure de créer un radar de recul fiable et précis qui vous aidera à garer votre voiture en toute simplicité.
- En suivant les instructions pas à pas, vous apprendrez à câbler les composants, à programmer la carte Raspberry Pi Pico et à tester le système pour vous assurer qu'il fonctionne correctement. Vous découvrirez également la physique des capteurs à ultrasons et la manière dont ils peuvent être utilisés pour mesurer la distance.

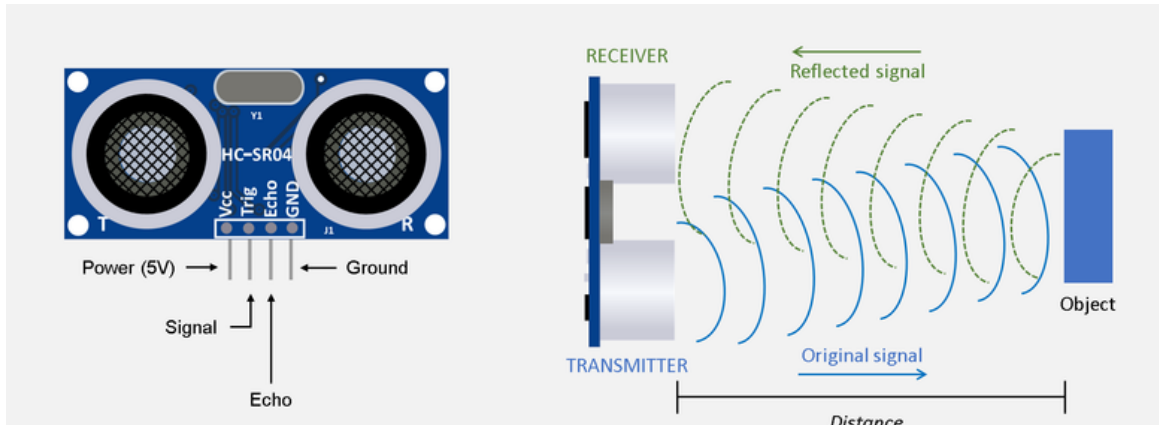
Les principes de fonctionnement du capteur à ultrasons HC-SR04 ?

Le capteur ultrasonique HC-SR04 fonctionne en émettant des ondes sonores à haute fréquence inaudibles à l'oreille humaine. Ces ondes sonores se déplacent dans l'air et rebondissent sur les objets qui se trouvent sur leur chemin. Lorsque les ondes sonores touchent un objet, elles sont réfléchies vers le capteur, qui mesure le temps que mettent les ondes à rebondir. En connaissant la vitesse du son et le temps nécessaire aux ondes pour se rendre jusqu'à l'objet et en revenir, le capteur peut calculer la distance qui le sépare de l'objet.

Voici les principales étapes des principes de fonctionnement du capteur ultrasonique HC-SR04 :

- Le capteur envoie une onde sonore à haute fréquence (généralement autour de 40 kHz) à partir de son émetteur.
- L'onde sonore se propage dans l'air et rebondit sur un objet.
- L'onde sonore réfléchie est détectée par le récepteur du capteur.

- Le capteur mesure le temps nécessaire à l'onde sonore pour se rendre à l'objet et en revenir.
- Le capteur calcule la distance qui le sépare de l'objet en fonction du temps que l'onde sonore a mis pour aller jusqu'à l'objet et en revenir.



Dans l'ensemble, le capteur ultrasonique HC-SR04 est un moyen fiable et précis de mesurer les distances, et il est couramment utilisé dans des applications telles que la robotique, l'automatisation et la détection de distance.

Objectifs :

Cette activité permettra à l'utilisateur de construire un système de radar de recul à l'aide de la carte Raspberry Pi Pico et d'un capteur ultrasonique HC-SR04. L'utilisateur acquerra des connaissances sur

- La physique des ondes ultrasoniques et comment elles peuvent être utilisées pour mesurer la distance.
- Les bases de la programmation en Python et comment écrire du code pour contrôler la carte Raspberry Pi Pico.
- Les principes de la conception de circuits et la manière de câbler des composants sur une carte de prototypage rapide pour créer un système de radar de recul fonctionnel.

En réalisant ce projet, l'utilisateur acquerra une meilleure compréhension de l'électronique, de l'ingénierie et de la programmation. Il disposera également d'un dispositif pratique et utile qu'il pourra utiliser pour rendre le stationnement de sa voiture plus sûr et plus commode.

Matériel à utiliser :

- 1 x Raspberry Pi Pico
- 1 x kit de carte à pain Pico

- 1 x Planche à pain de taille normale
- 1 x Capteur ultrasonique HC-SR04
- 1 x Buzzer
- 3 x LED (verte, jaune et rouge)
- 3 x résistances de 220 ohms
- Fils de connexion

Étapes à suivre :

Les principales étapes pour réaliser l'expérience du radar de recul avec la carte Raspberry Pi Pico et le capteur ultrasonique HC-SR04 :

1. Connecter le capteur ultrasonique HC-SR04 à la carte Raspberry Pi Pico à l'aide de fils de connexion.
2. Connecter le buzzer et les LED à la carte Raspberry Pi Pico à l'aide de fils de connexion et de résistances de 220 ohms pour limiter le flux de courant.
3. Écrire un programme Python pour contrôler la carte Raspberry Pi Pico et utiliser le capteur HC-SR04 pour mesurer les distances.
4. Programmer la carte Raspberry Pi Pico pour qu'elle allume la LED verte lorsqu'il n'y a pas d'obstacle, la LED jaune lorsque l'obstacle se trouve dans une certaine zone et la LED rouge lorsque l'obstacle est trop proche.
5. Programmer la carte Raspberry Pi Pico pour activer le buzzer lorsque l'obstacle est trop proche.
6. Tester le système de radar de recul en plaçant des obstacles à différentes distances et à différents angles derrière le radar et s'assurer que les LED et l'avertisseur sonore fournissent le retour d'information approprié.

Pour l'étape 4 : Pour étudier le fonctionnement du système, il est nécessaire de programmer son comportement. Les valeurs de distance peuvent être ajustées en fonction des besoins spécifiques. La programmation doit suivre les règles suivantes :

- La LED verte doit être allumée en permanence si la distance entre le capteur et un obstacle est supérieure à 20 cm (1 mètre).
- La LED orange doit être allumée si la distance entre le capteur et un obstacle est comprise entre 20 cm et 5 cm.
- La LED rouge doit être allumée si la distance entre le capteur et un obstacle est inférieure à 5 cm.

En programmant le système pour qu'il suive ces règles, il fournira un retour visuel clair de la distance entre le capteur et tout obstacle.

Ces règles peuvent être adaptées en fonction des besoins ou des cas d'utilisation spécifiques.

Schéma de câblage

Voici un schéma de câblage pour le projet de radar de recul avec la carte Raspberry Pi Pico, le capteur ultrasonique HC-SR04, le buzzer et trois LED :

Carte Raspberry Pi Pico :

- GP15 : Broche de déclenchement du capteur HC-SR04
- GP14 : Broche d'écho du capteur HC-SR04
- GP10 : Broche positive de la LED verte
- GP11 : Broche positive de la LED orange
- GP12 : Broche positive de la LED rouge
- GP2 : Broche positive du buzzer
- GND : Broche de mise à la terre de la carte

Capteur HC-SR04 :

- VCC : Connexion à une source d'alimentation de 5V
- GND : Connexion à la masse de la carte Raspberry Pi Pico
- Trig : Connecter à GP15 de la carte Raspberry Pi Pico
- Echo : Connecter au GP14 de la carte Raspberry Pi Pico

LED verte :

- Branche positive : Connecter au GP10 de la carte Raspberry Pi Pico via une résistance de 220 ohms.
- Branche négative : Connecter à GND de la carte Raspberry Pi Pico

LED orange :

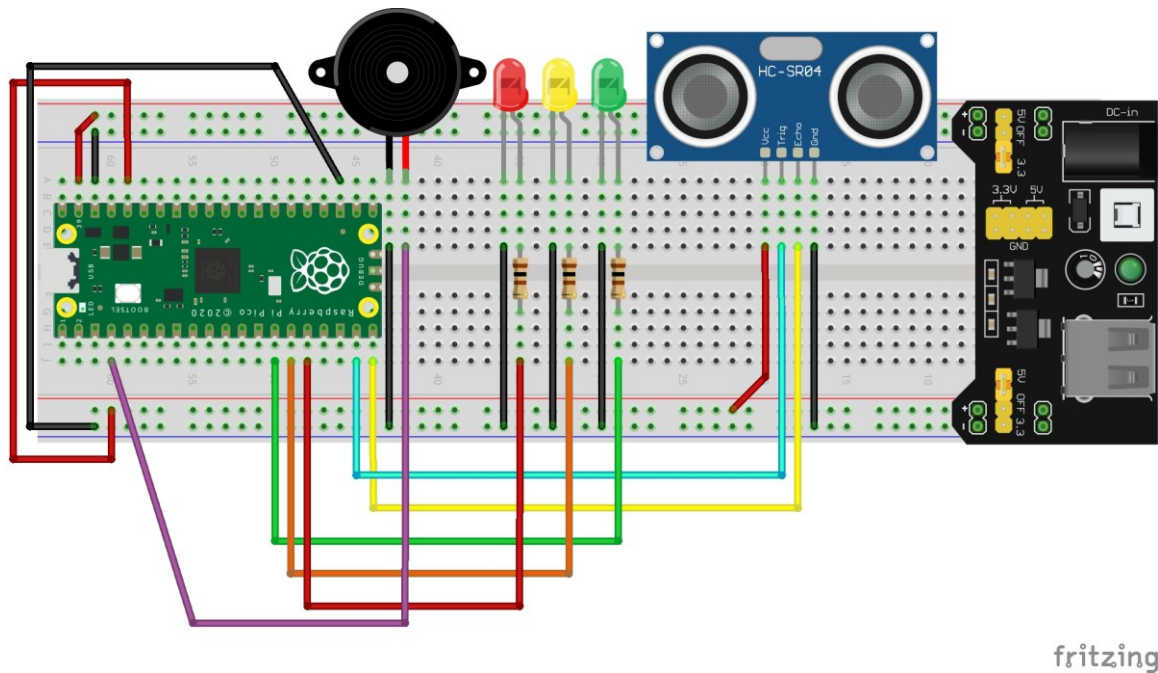
- Branche positive : Connecter à GP11 de la carte Raspberry Pi Pico via une résistance de 220 ohms.
- Branche négative : Connecter à la masse (GND) de la carte Raspberry Pi Pico

LED rouge :

- Branche positive : Connecter à GP12 de la carte Raspberry Pi Pico via une résistance de 220 ohms.
- Branche négative : Connecter à la masse (GND) de la carte Raspberry Pi Pico

Buzzer :

- Branche positive : Connecter à GP2 de la carte Raspberry Pi Pico
- Branche négative : Connecter à GND de la carte Raspberry Pi Pico



Code

```
# Import required libraries
from machine import Pin, time_pulse_us
import time

# Define pins for the components
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)

# Define the distance thresholds for each LED
green_threshold = 20 # cm
orange_threshold = 5 # cm

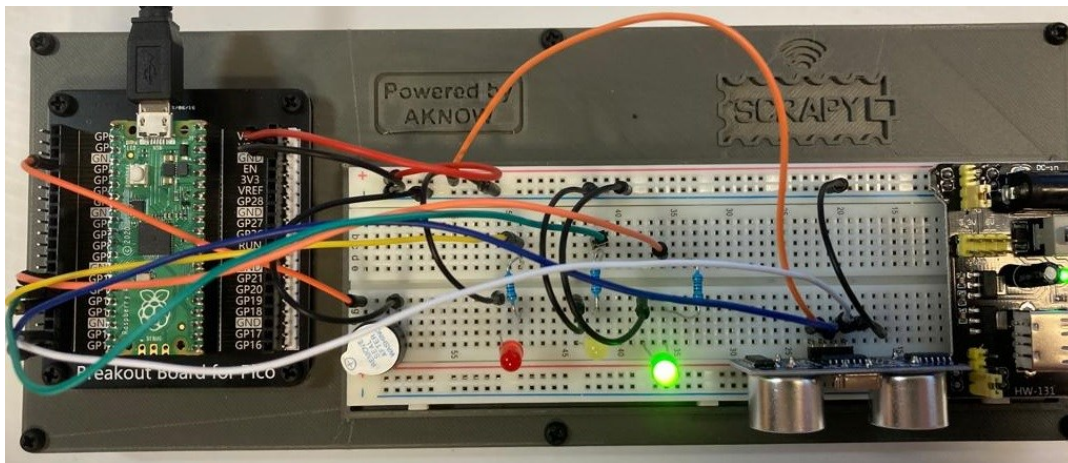
# Define the function to calculate the distance from the HC-SR04 sensor
def get_distance():
    # Send a 10us pulse to trigger the sensor
    trigger_pin.low()
    time.sleep_us(2)
    trigger_pin.high()
    time.sleep_us(10)
    trigger_pin.low()
    # Measure the duration of the echo signal
    duration = time_pulse_us(echo_pin, 1, 10000)
    # Calculate the distance from the duration using the speed of sound (343
    # m/s)
    distance = duration / 2 / 1000000 * 343 * 100
    return distance

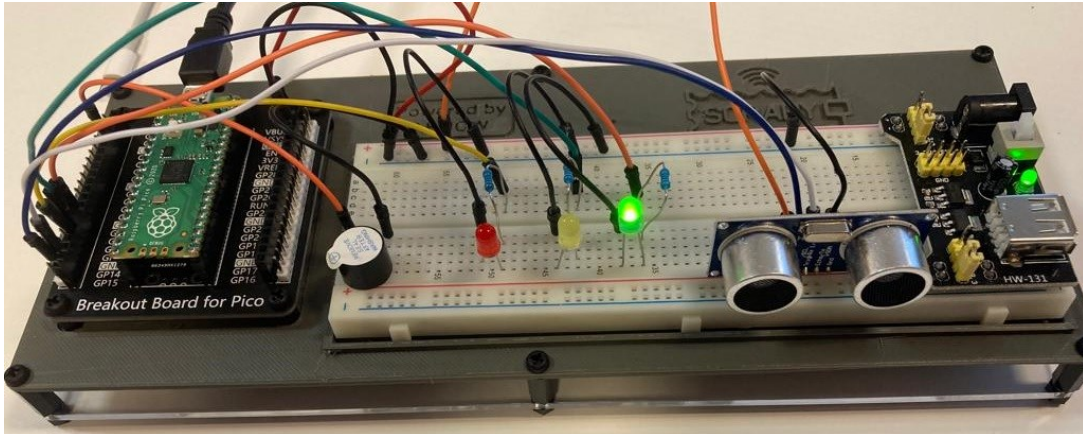
# Define the main loop to read the distance and control the LEDs and buzzer
while True:
```



```
# Get the distance from the sensor and print it
distance = get_distance()
print(f"Distance : {distance} cm")
# Turn on the green LED if the distance is greater than the threshold
if distance > green_threshold:
    green_led.on()
    orange_led.off()
    red_led.off()
    buzzer.off()
# Turn on the orange LED if the distance is between the thresholds
elif distance > orange_threshold:
    green_led.off()
    orange_led.on()
    red_led.off()
    buzzer.off()
# Turn on the red LED and buzzer if the distance is less than the threshold
else:
    green_led.off()
    orange_led.off()
    red_led.on()
    buzzer.on()
    time.sleep(0.5) # Buzz for 0.5 seconds
# Wait for 0.1 second before the next measurement
time.sleep(0.1)
```

Photos





Conclusion

En conclusion, ce projet consistait à utiliser une carte Raspberry Pi Pico, un capteur ultrasonique HC-SR04 et quelques composants supplémentaires pour créer un radar de recul pour les voitures. Nous avons appris les principes de fonctionnement du capteur HC-SR04, comment câbler les composants ensemble et comment programmer le comportement du système à l'aide de MicroPython. En suivant les étapes de ce tutoriel, nous avons pu créer un système capable de détecter les obstacles et de fournir un retour visuel et sonore au conducteur.

En ce qui concerne la poursuite de l'exploration, il existe de nombreuses façons d'étendre ce projet. Voici quelques suggestions :

- Ajouter plus de LED ou un écran pour fournir un retour d'information plus détaillé sur la distance.
- Utiliser des algorithmes d'apprentissage automatique pour améliorer la détection et la précision des obstacles.
- Créer une interface sans fil pour permettre au système de communiquer avec un appareil mobile ou un autre appareil externe.
- Incorporer des capteurs ou des composants supplémentaires pour créer un système de sécurité automobile plus complet.

Dans l'ensemble, ce projet constitue une excellente introduction au monde de l'électronique et de la programmation et sert de point de départ à d'autres explorations et expérimentations.