



PREMIERE GENERALE

NUMERIQUE ET SCIENCES INFORMATIQUES

MINI-PROJET : SYSTEME D'AIDE AU STATIONNEMENT ARRIERE

1 – SYSTEME D'AIDE AU STATIONNEMENT ARRIERE

Le dispositif d'aide au stationnement arrière ou radar de recul (en anglais : Parking aids rear) est un dispositif qui permet au conducteur de se garer plus facilement et en toute sécurité. Constitué d'au moins 4 capteurs disposés sur les pare-chocs arrières il permet de signaler la présence d'obstacles derrière le véhicule lors de la marche arrière et ainsi préserver l'intégrité de la voiture, mais également celle d'un éventuel piéton ou enfant hors du champ de vision.



Lors d'une manoeuvre en marche arrière, le conducteur est informé par un bip dont le rythme d'émission s'accélère au fur et à mesure que le véhicule se rapproche d'un éventuel obstacle. Une information visuelle peut venir en complémentarité du dispositif d'avertissement sonore. L'indicateur visuel est constitué de LED ou d'un bargraphe qui augmente au fur et à mesure que le véhicule se rapproche de l'obstacle.



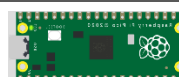
2 – MINI-PROJET

2.1 – Présentation du mini-projet

Le mini-projet consiste à mettre en œuvre un système qui permette la **mesure de la distance** avec un éventuel obstacle, l'affichage d'une **information visuelle** et d'une **information sonore** qui indique la distance avec l'obstacle.

2.2 – Matériel disponible

Une carte Raspberry PICO.



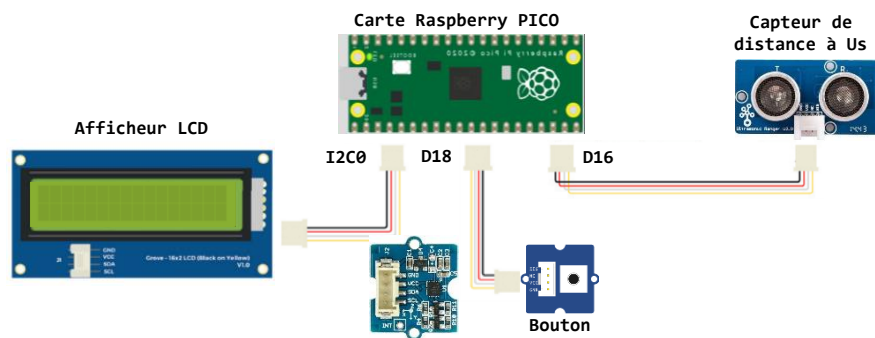
Capteur de distance à ultrasons grove.



Afficheur LCD I2C Grove



2.3 – Câblage des composants



2.4 – Cahier des charges

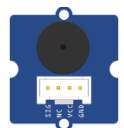
- Le système devient actif lorsque la marche arrière est enclenchée. L'enclenchement de la marche arrière sera simulée par un appui sur un bouton. Le désenclenchement sera effectif après un nouvel appui sur ce bouton.
- Le capteur de distance doit permettre de mesurer une distance comprise entre 30 cm et 150 cm.
- La distance mesurée sera affichée sur l'écran LCD.
- Si la distance est inférieure à 30 cm un message d'alerte sera affiché sur l'écran LCD.
- Pour aller plus loin Une information visuelle sera réalisée d'une LED RGB.



Obstacle à moins de 10 cm	Obstacle entre 10 et 50 cm	Obstacle entre 50 et 150 cm	Obstacle à plus de 150 cm
LED allumée rouge	LED allumée jaune	LED allumée verte	LED éteinte

- Pour aller plus loin : Un avertissement sonore sera émis par un module buzzer grove :

Distance	Avertissement
$D > 150 \text{ cm}$	Pas d'avertissement
$30 \text{ cm} < D < 150 \text{ cm}$	Bip sonore dont la fréquence augmente avec la diminution de la distance
$D < 30 \text{ cm}$	Bip continu



3 – COMPOSANTS

3.1 – Capteur de distance à ultrasons

Le capteur de distance à ultrasons grove est un module à ultrasons qui permet de mesurer des distances comprises entre 2 cm et 350 cm avec une tolérance de $\pm 1 \text{ cm}$. Il est constitué d'un émetteur et récepteur ultrason. La fréquence des ultrasons utilisée est de 40 kHz.

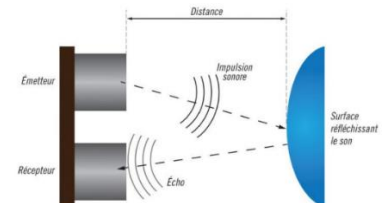


Caractéristiques techniques :

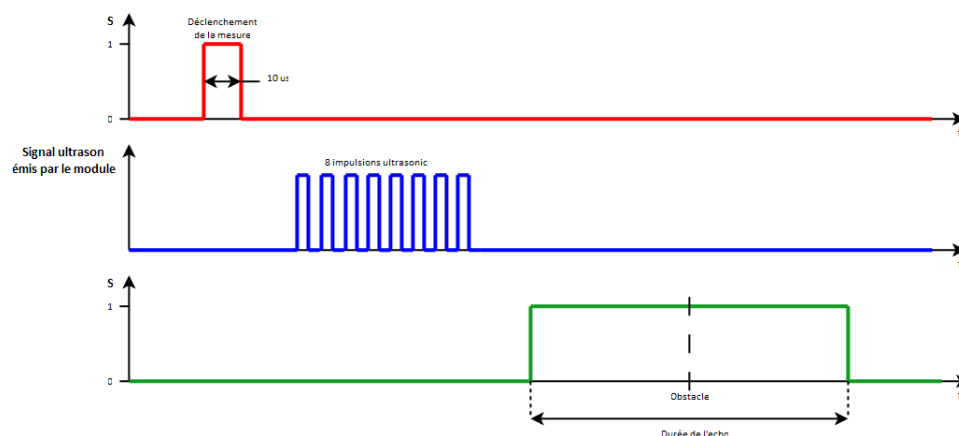
Caractéristiques	Valeurs		Unité
Tension d'alimentation	Min	3.2	V
	Max	5.2	
Courant		8	mA
Fréquence		40	kHz
Plage de mesure	Min	2	cm
	Max	350	
Résolution		1	cm

Principe de fonctionnement :

Le principe de mesure consiste à envoyer une salve ultrasonore par la partie émettrice puis de mesurer le temps mis par la salve pour se réfléchir sur l'obstacle et revenir vers le récepteur du capteur à ultrasons. Connaissant la durée de l'aller et du retour et la vitesse de propagation du son, on en déduit alors la distance.



Pour déclencher la mesure, il faut générer une **impulsion à l'état haut** (niveau logique 1) sur la broche SIG du module pendant au moins 10 μ s. Dès que le signal ultrason est émis, le module place la broche S à l'état haut jusqu'à ce que l'écho soit capté par le récepteur ultrason. Il suffit alors de **mesurer la durée pendant laquelle la broche S est placée à l'état haut**.



L'algorithme de la fonction permettant de mesurer la distance avec un obstacle est le suivant :

```

VARIABLES
    duree : nombre entier
    distance : nombre réel
DEBUT
    Placer la broche SIG en sortie
    Placer la broche SIG à l'état haut
    Attendre 10 us
    Placer la broche SIG à l'état bas
    Placer la broche SIG en entrée
    Lire la durée de l'état haut sur la broche SIG
    Calculer la distance
FIN
  
```

Pour mesurer la durée d'une impulsion, il faut utiliser la fonction `time_pulse_us(nom_broche, Etat_impulsion)` contenue dans le module `machine`. La fonction `time_pulse_us(nom_broche, Etat_impulsion)` retourne la durée, en microsecondes, d'une impulsion à l'état bas ou l'état haut selon la valeur du paramètre `Etat_impulsion`, sur la broche spécifiée par le paramètre `nom_broche`.

Brochage



Broche	Nom	Câble	Rôle
1	SIG	Jaune	Sortie analogique
2	NC	Blanc	Non connectée
3	VCC	Rouge	Alimentation
4	GND	Noir	Masse (0V°)

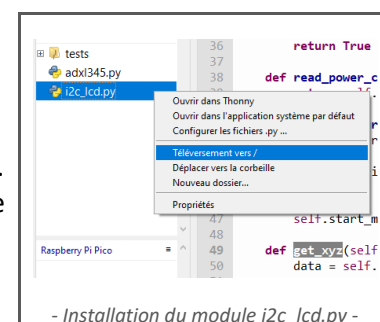
3.2 – Afficheur LCD

L'afficheur LCD doit être connecté sur le bus I2C de la carte Raspberry PICO. Il permet un affichage de 2 lignes de 16 caractères blancs sur un rétroéclairage bleu.



Le module, **i2c_lcd** permet d'utiliser cet afficheur.

Pour pouvoir utiliser ce module il faut le téléverser dans la carte Raspberry PICO. Pour cela rechercher, dans la zone « Fichiers » le fichier **i2c_lcd.py** puis avec le bouton droit de la souris cliquer sur « Téléversement vers/ ».



- Installation du module i2c_lcd.py -

L'accéléromètre communique avec une liaison I2C. Il est donc nécessaire de créer un objet **I2C** en spécifiant les broches utilisées pour l'interface I2C et éventuellement la fréquence de la liaison. Il faut ensuite créer un objet **lcd** qui nous permettra de gérer l'afficheur.

```
from machine import Pin, I2C
from i2c_lcd import lcd

i2c = I2C(0, scl = Pin(9), sda = Pin(8), freq = 400000)
display = lcd(i2c)
```

Les principales méthodes de ce module sont :

Methode	Rôle	Exemple
lcd()	Création d'un objet de la classe lcd permettant de piloter l'afficheur	display = lcd()
write_char(c)	Affichage du caractère c	display.write_char('T')
write(chaine)	Affichage la chaine de caractères chaine	display.write('Temp')
cursor(s)	Curseur visible si s = 1 (par défaut) et invisible si s = 0	display.cursor(0)
setCursor(x, y)	Placement du curseur sur la colonne x et la ligne y	display.setCursor(0, 1)
display(s)	Affichage visible si s = 1 (par défaut) et invisible si s = 0	display.display(0)
clear()	Permet d'effacer l'affichage	display.clear()
home()	Placement du curseur sur la colonne 0 et la ligne 0	display.home()

Brochage



Broche	Nom	Câble	Rôle
1	SCL	Jaune	Ligne Horloge liaison I2C
2	SDA	Blanc	Ligne Données liaison I2C
3	VCC	Rouge	Alimentation
4	GND	Noir	Masse (0V°)