




















Perception de l'espace de la voiture autonome

Matthieu Castetz n°47995



LES 6 NIVEAUX D'AUTONOMIE D'UN VÉHICULE

	ACCÉLÉRATION FREINAGE & VOLANT	SURVEILLANCE DE LA ROUTE	CONTRÔLE EN CAS DE PROBLÈME	QUELLES CONDITIONS
Niveau 0				
Niveau 1				Certaines routes
Niveau 2				Certaines routes
Niveau 3				Certaines routes
Niveau 4				Certaines routes
Niveau 5				Toutes les routes

Niveau 3 autorisé le 1^{er} septembre 2022

- ▶ 1 Principe de la détection par Ultrasons
- ▶ 2 Montage d'une voiture autonome
- ▶ 3 Programmation Arduino
- ▶ 4 Limites de la méthode

HC SR04

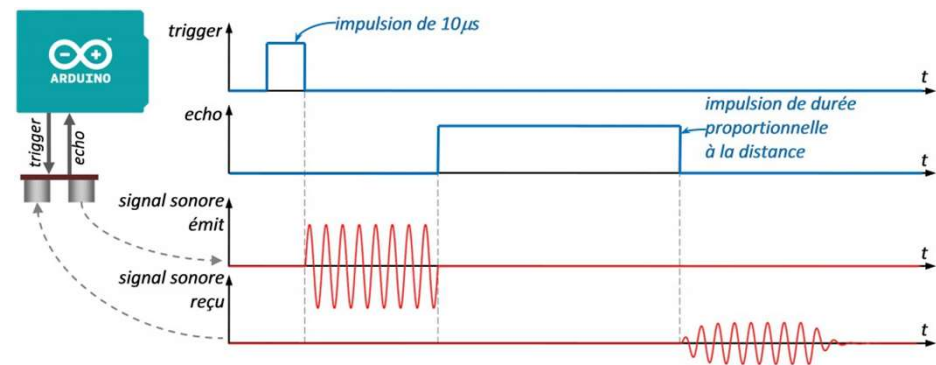
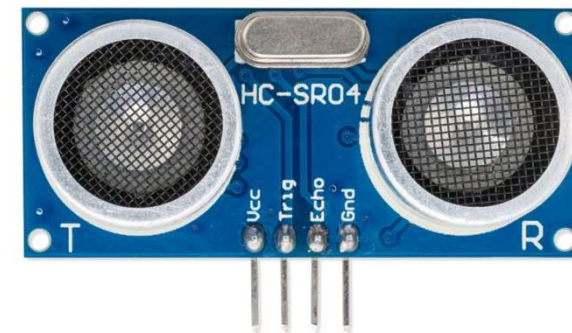
Plage de mesure: 2 cm à 400 cm

Angle de mesure efficace: 15°

Prix $\approx 1,50\text{€}$

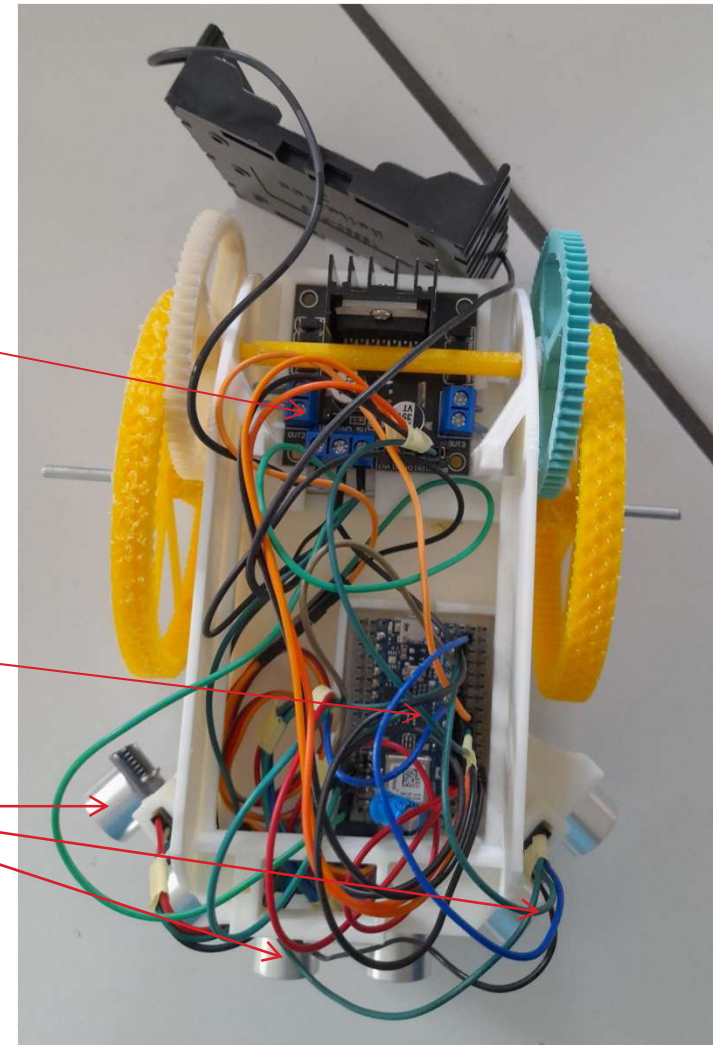
Émet a $f=40\text{Hz}$

Fonctionne comme un sonar



Double pont en H
relié à 2 moteurs
entraînant des roues dentées

Arduino nano 33ble
sur une plaque labdec
3 Capteurs ultrasons

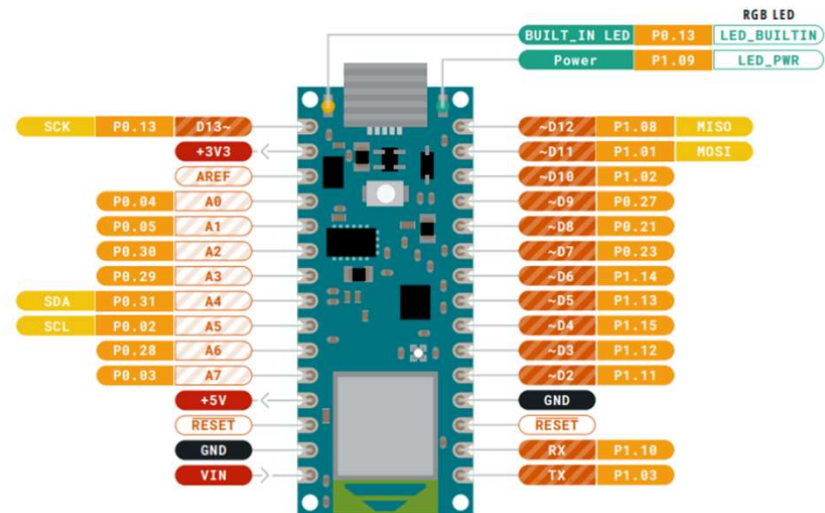
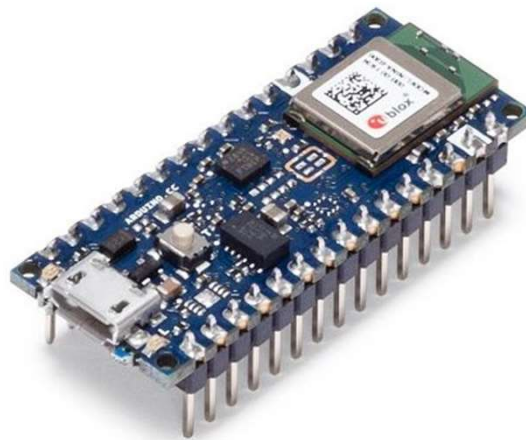


Alimentation à 3,3V

14 pins digitals

8 pins analogiques

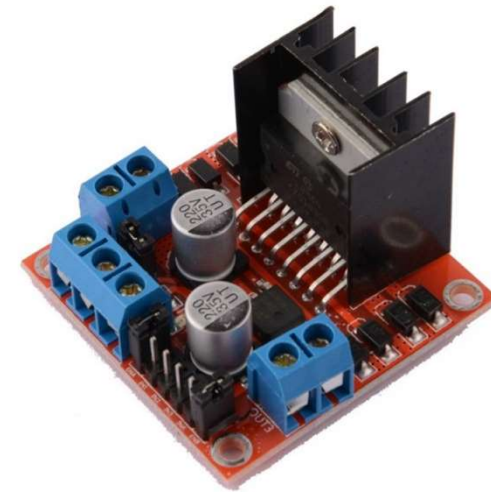
5 pins PWM



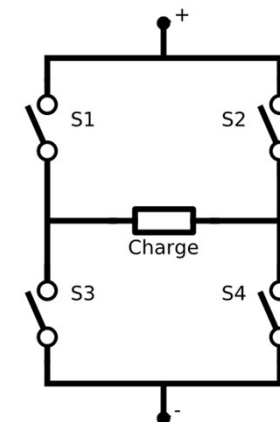
L298N

Prix $\approx 2,70\text{€}$

Relié à deux moteurs DC 1.5V



Double pont en H



Servomoteur 9g

Rotation entre 0° et 180°

Poids de 9 grammes

Puissance suffisante pour lever
une petite barrière métallique

Prix:3,60€




```
void loop() {  
  // Activer les moteurs pour avancer  
  motorForward();  
  
  // Vérifier la présence d'obstacles avec les émetteurs/récepteurs ultrasons  
  bool obstacle_avant = detecterObstacle_avant();  
  bool obstacle_gauche = detecterObstacle_gauche();  
  bool obstacle_droite = detecterObstacle_droite();  
  
  // Si un obstacle est détecté par un émetteur, tourner pour l'éviter avec le servomoteur  
  if (obstacle_avant || obstacle_gauche || obstacle_droite) {  
    // Tourner pour éviter l'obstacle avec le servomoteur  
    eviterObstacle(obstacle_avant, obstacle_gauche, obstacle_droite);  
    pivoterToutDroit();  
  }  
  
  // Si les trois émetteurs détectent un obstacle, effectuer un demi-tour  
  if (obstacle_avant && obstacle_gauche && obstacle_droite) {  
    demi_tour();  
  }  
}
```

```
bool detecterObstacle_avant() {  
    // Envoi de la décharge de déclenchement  
    digitalWrite(trigPin_avant, LOW);  
    delayMicroseconds(2);  
  
    // Déclenchement de la mesure en envoyant une impulsion de 10 µs  
    digitalWrite(trigPin_avant, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigPin_avant, LOW);  
  
    // Lecture de la durée de l'impulsion écho  
    long duration = pulseIn(echoPin_avant, HIGH);  
  
    // Calcul de la distance en centimètres  
    int distance_cm = duration / 58;  
  
    // Vérification de la distance détectée  
    if (distance_cm < 30) {  
        return false; // Obstacle détecté  
    } else {  
        return true; // Pas d'obstacle détecté  
    }  
}
```

```
// Fonction pour tourner le servomoteur pour éviter l'obstacle
void eviterObstacle(bool obstacle_avant, bool obstacle_gauche, bool obstacle_droite) {
    // Vérifier la position de l'obstacle pour décider la direction à tourner
    if (obstacle_avant && !obstacle_gauche && !obstacle_droite) {
        // Tourner à gauche
        pivoterGauche();
    } else if (!obstacle_avant && obstacle_gauche && !obstacle_droite) {
        // Tourner à droite
        pivoterDroite();
    } else if (!obstacle_avant && !obstacle_gauche && obstacle_droite) {
        // Tourner à gauche
        pivoterGauche();
    }
}
```

Détection Us

Montage

Arduino

Limites

```
// Fonction pour pivoter à gauche  
void pivoterGauche() {  
  servo.write(120);  
}
```

```
// Fonction pour activer les 2 moteurs en avant  
void motorForward() {  
  digitalWrite(in1Pin, HIGH);  
  digitalWrite(in2Pin, LOW);  
  digitalWrite(in3Pin, HIGH);  
  digitalWrite(in4Pin, LOW);  
}
```

Fonctions d'orientation et de déplacement

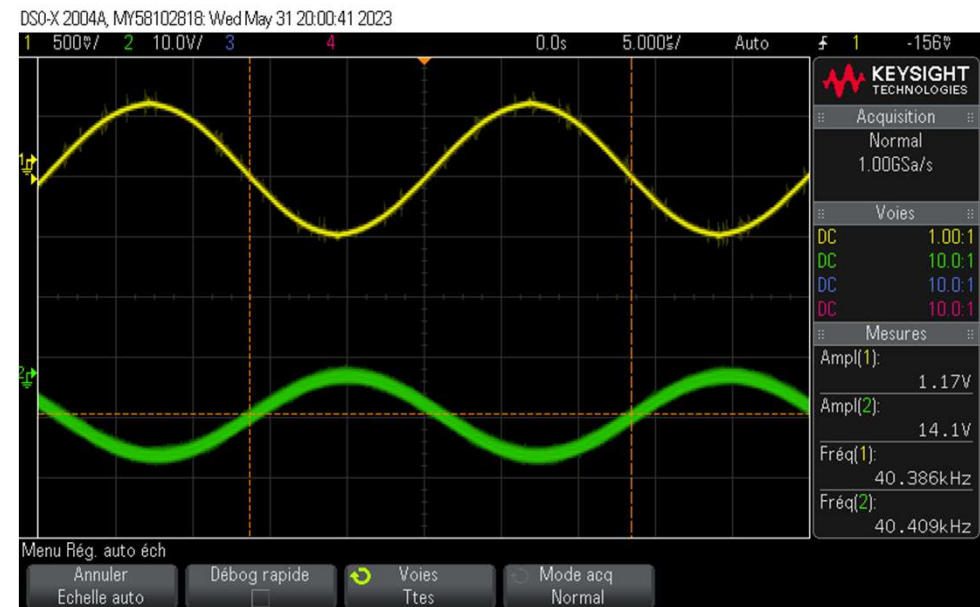
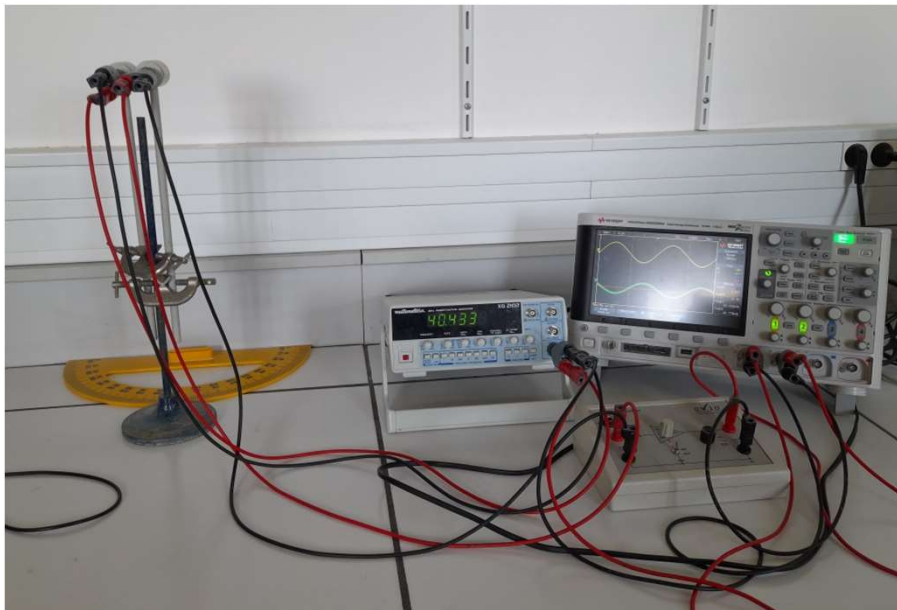
Détection Us

Montage

Arduino

Limites

Absorption des ondes US

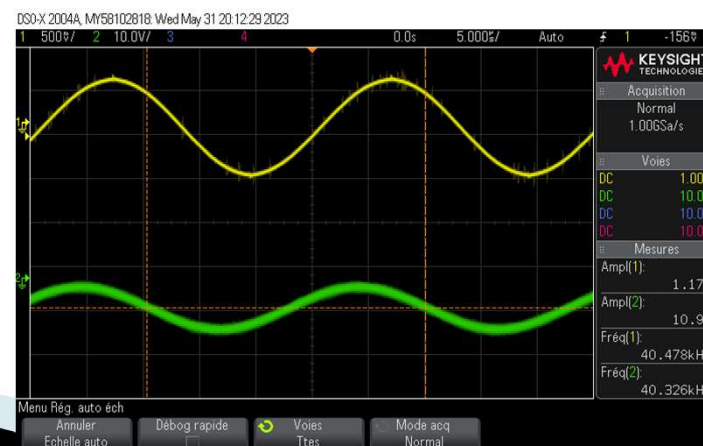
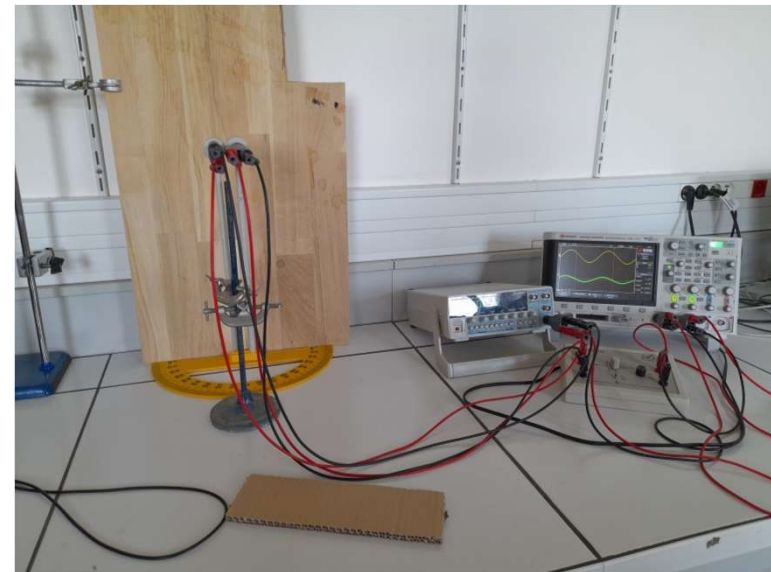
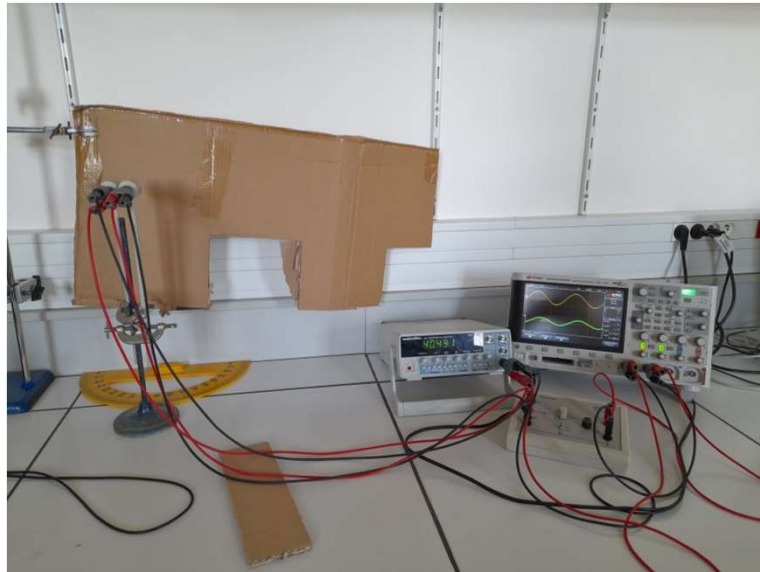


Détection Us

Montage

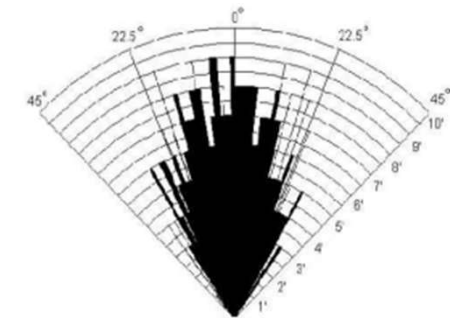
Arduino

Limites

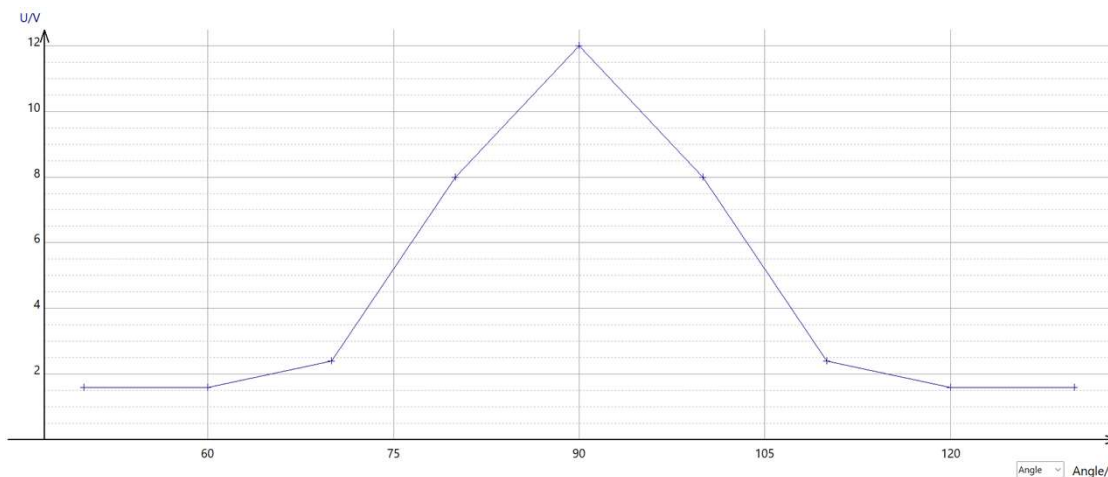


Angle de vue limité

L'orientation des obstacles peut poser problème



Practical test of performance,
Best in 30 degree angle



i	Angle	U
	°	V
0	50,00	1,600
1	60,00	1,600
2	70,00	2,400
3	80,00	8,000
4	90,00	12,00
5	100,0	8,000
6	110,0	2,400
7	120,0	1,600
8	130,0	1,600
9		