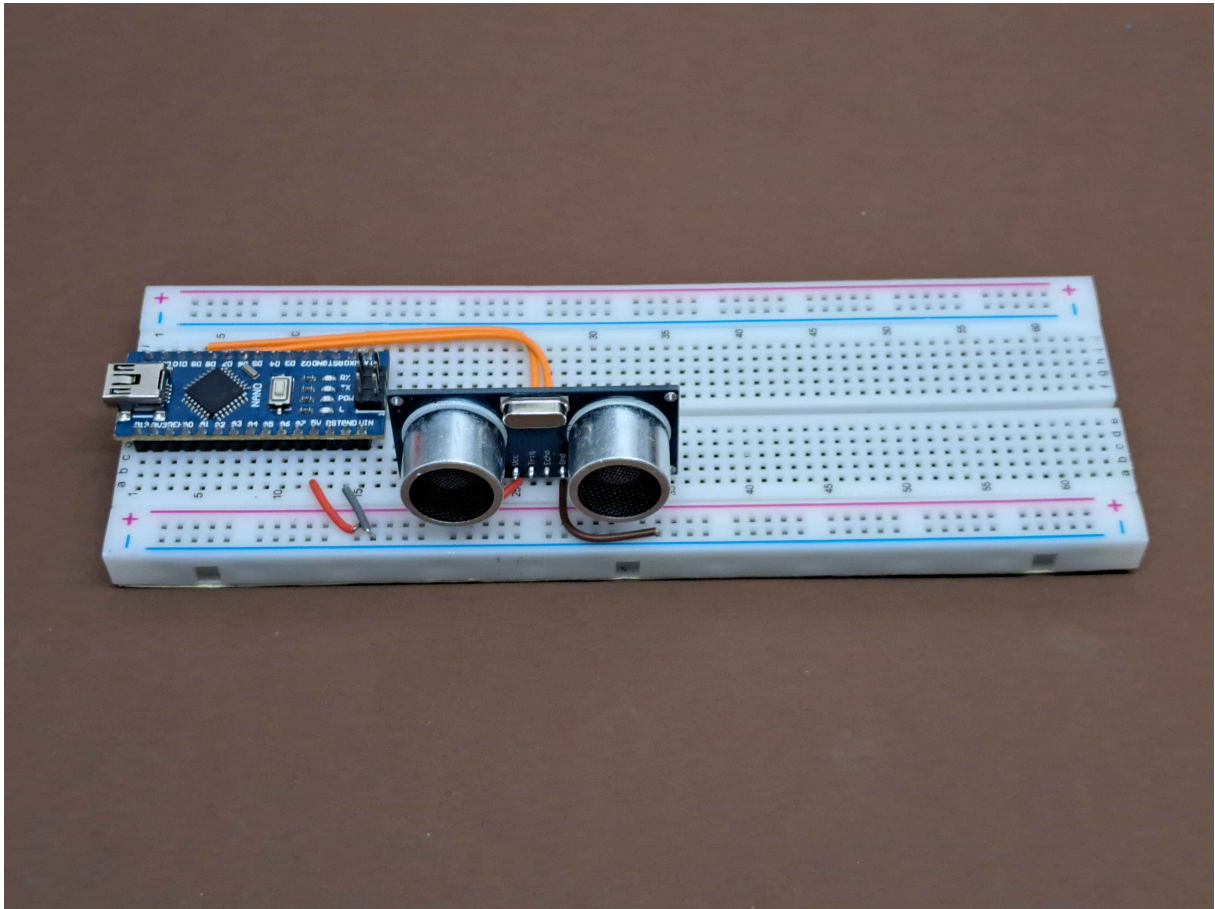


Atelier : Calcul de Distance avec le Capteur Ultrason HC-SR04



Circuit final

Étape 1 : Introduction au HC-SR04

Le capteur ultrason HC-SR04 vous permet de mesurer la distance à l'aide d'ondes sonores. C'est un outil idéal pour les projets impliquant la détection d'obstacles ou la mesure de distances précises.

Écho de l'ultrason

Caractéristiques

- Portée de $2cm$ à $400cm$
- Précision : $\pm 3mm$
- Angle de détection : $\approx 15^\circ$

Matériel

- Arduino Uno
- Capteur Ultrason HC-SR04
- Câbles de connexion

Étape 2 : Connexions du HC-SR04

Pour utiliser efficacement le HC-SR04, vous devez connecter les broches VCC, Trig, Echo, et GND à votre Arduino. Voici un aperçu des connexions.

Schéma de Connexion

1. **VCC** → Reliez à la broche 5V.
2. **Trig** → Choisissez une broche digitale, par exemple, 9.
3. **Echo** → Optez pour une broche digitale comme 10.
4. **GND** → Connectez au GND de l'Arduino.

Étape 3 : Programmation de Base

Configuration

1. Déclarez les broches pour le HC-SR04 en utilisant des macros pour `Trig` et `Echo` :

```
1 #define PIN_TRIG 9
2 #define PIN_ECHO 10
```

2. Initialisez ces broches dans la fonction `setup()` :

```
1 pinMode(PIN_TRIG, OUTPUT);
2 pinMode(PIN_ECHO, INPUT);
```

Boucle Principale

Afin de mesurer la distance, vous devez envoyer un signal **HIGH** de quelques *microsecondes* sur la broche **Trig**. Le HC-SR04 enverra ensuite un signal ultrasonique et attendra le retour du signal sur la broche **Echo**.

Nous allons ensuite mesurer la durée du signal **HIGH** sur la broche **Echo** pour calculer la durée de l'écho.

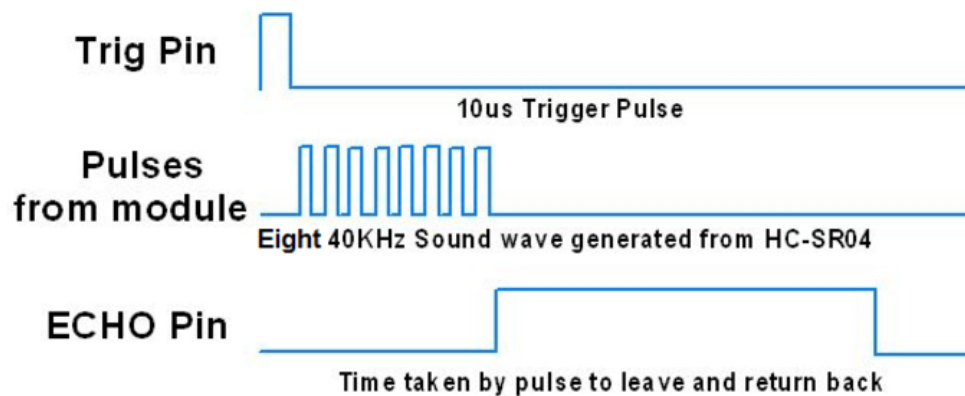


Figure 1: Signaux du HC-SR04 en fonction du temps

1. Envoyez un signal court sur **Trig** pour initier une mesure.

```
1 digitalWrite(PIN_TRIG, LOW);  
2 delayMicroseconds(2);  
3 digitalWrite(PIN_TRIG, HIGH);  
4 delayMicroseconds(10);  
5 digitalWrite(PIN_TRIG, LOW);
```

2. Utilisez `pulseIn` pour capturer la durée du signal **HIGH** sur **Echo**.
3. Calculez la distance à partir du temps mesuré :

```
1 int distance = duration / 29.1 / 2;
```

4. Affichez cette distance sur le moniteur série.

Pourquoi 29.1 ?

La vitesse du son dans l'air est d'environ 29.1 cm/s . Nous divisons par la vitesse du son en cm/s pour ensuite diviser par 2 pour tenir compte de l'aller-retour du signal.

Étape 4 : Créer une Fonction pour lire la Distance

Pour simplifier votre code, créez une fonction qui retourne la distance mesurée par le HC-SR04.

Fonction `getDistance()`

1. Déclarez une fonction qui renvoie un entier :

```
1 int getDistance() {  
2     // Code de mesure de distance  
3 }
```

2. Copiez le code de mesure de distance dans cette fonction.
3. Remplacez `int distance` par `return distance`.

Utilisation

Appelez cette fonction dans la boucle principale :

```
1  
2 void loop() {  
3     int distance = getDistance();  
4     Serial.print("Distance : ");  
5     Serial.print(distance);  
6     Serial.println(" cm");  
7     delay(1000);  
8 }
```

Étape 5 : Modéliser un cadre pour le Capteur HC-SR04

Pour rendre votre projet plus professionnel, vous pouvez imprimer un cadre pour le HC-SR04.

Vous pouvez utiliser [TinkerCAD](#) ou [OpenSCAD](#) pour modéliser un cadre simple qui maintient le HC-SR04 en place.

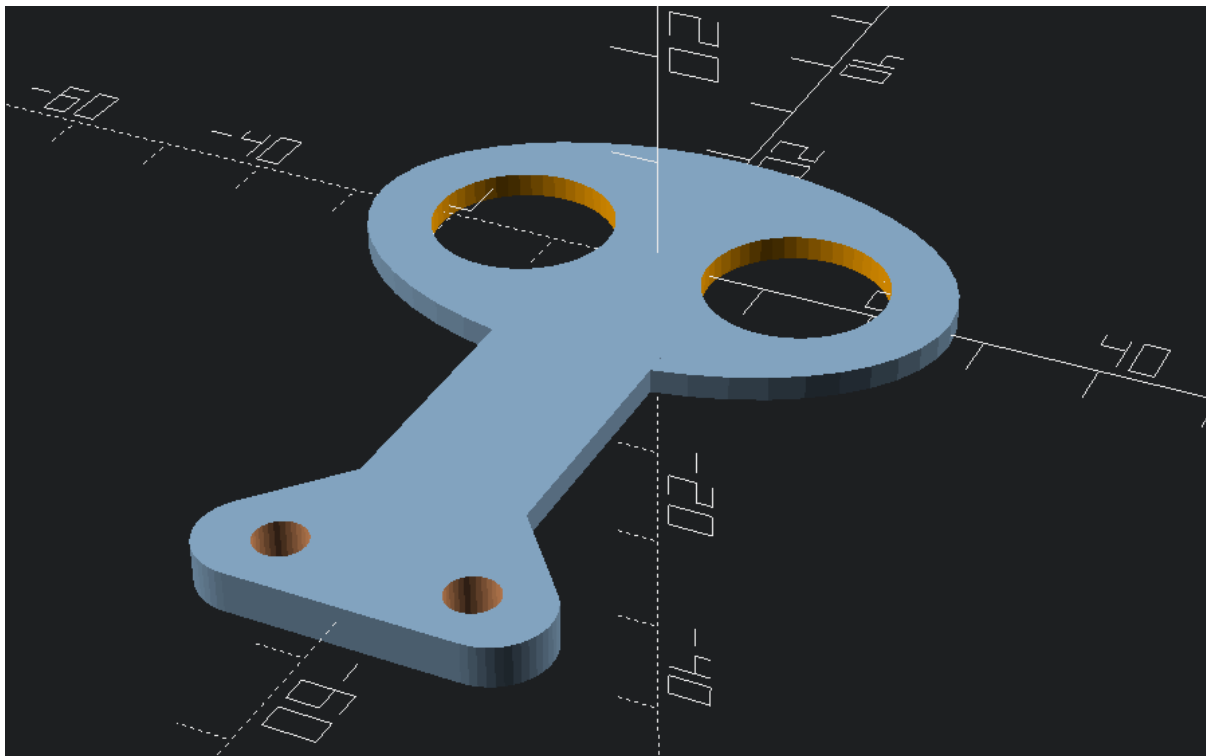


Figure 2: Exemple de cadre de HC-SR04

Voir la page suivante pour les mesures précises du cadre HC-SR04.

Mesures du HC-SR04

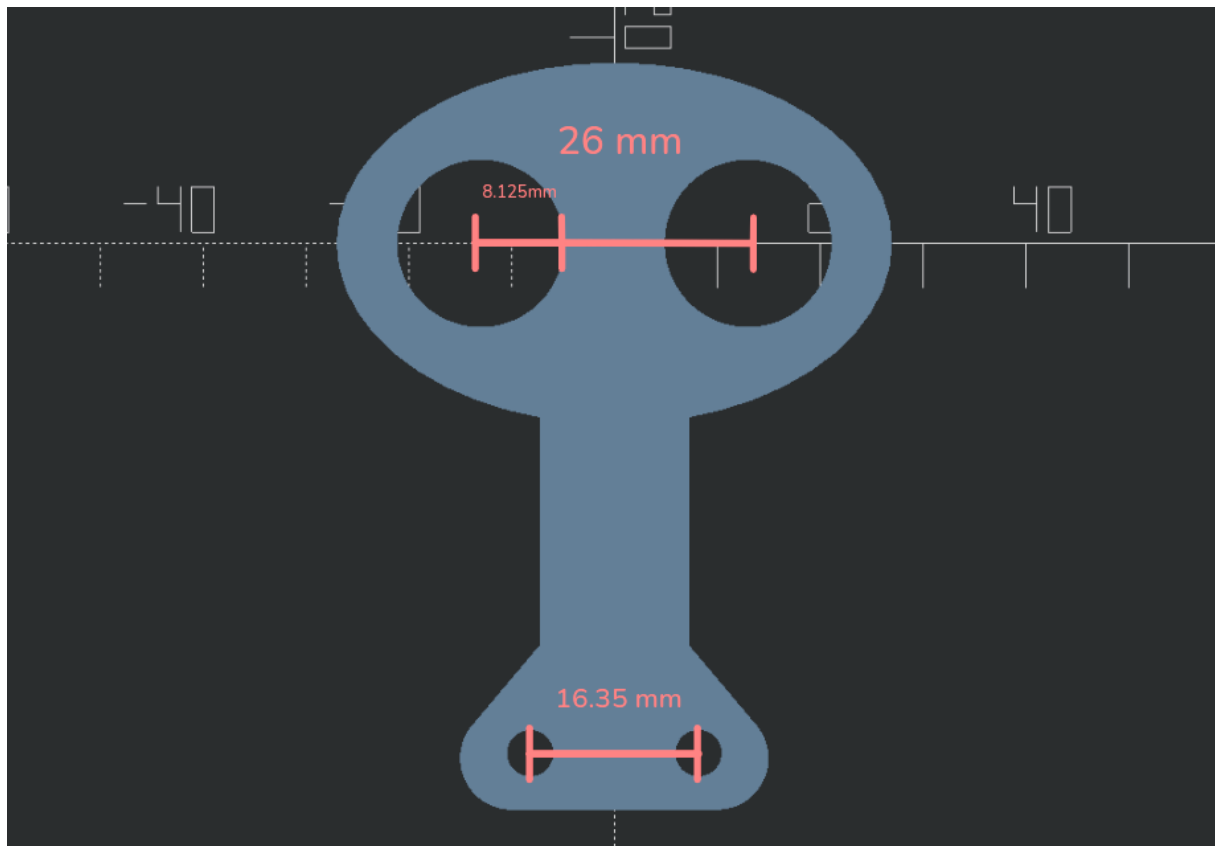


Figure 3: Mesures importantes du HC-SR04

- Épaisseur du cadre : $1.5mm$ à $3mm$
- Diamètre d'un oeil : $16.25mm$
- Distance entre les deux centres des yeux : $26mm$
- Diamètre des trous de fixation: $3.9mm$ à $4.1mm$
- Distance entre les trous de fixation : $16.35mm$

Exportation

1. Exportez votre modèle en format **.stl** pour l'impression 3D.
2. Envoyez le fichier **.stl** à votre professeur pour le "slicing" et l'impression.