

# Club Robotique

## Cycle 5 - Poubelle Connectée

- Phase 1 - Théorie et fonctionnement des composants
- Phase 2 - Utilisation des composants et programmation de la poubelle connectée

# Phase 1 - Théorie et fonctionnement des composants

## Phase 1 - Théorie et fonctionnement des composants

### Objectifs pédagogiques :

- Comprendre le fonctionnement de l'Arduino Nano et son Shield.
- Découvrir le rôle d'un capteur à ultrasons (HC-SR04).
- Apprendre comment fonctionne un servomoteur et comment le contrôler.
- Apprendre à câbler ces composants ensemble correctement pour un projet Arduino.

### Matériel nécessaire :

- 1 x **Arduino Nano**
  - 1 x **Shield pour Arduino Nano** (ou une breadboard pour faciliter les connexions)
  - 1 x **Capteur à ultrasons HC-SR04**
  - 1 x **Servomoteur SG90**
  - Fils de connexion
  - 1 x **Câble USB** pour programmer l'Arduino Nano
  - 1 x **Ordinateur** avec l'IDE Arduino installé
- 

## 1. Introduction à l'Arduino Nano

### Qu'est-ce que l'Arduino ?

<https://innovation.iha.unistra.fr/books/robotique-educative/page/quest-ce-que-larduino>

---

## 2. Capteur à ultrasons HC-SR04

## Comment fonctionne le capteur à ultrasons ?

<https://innovation.iha.unistra.fr/books/robotique-educative/page/quest-ce-quun-capteur-a-ultrasons>

---

## 3. Fonctionnement du servomoteur SG90

### Comment fonctionne un servomoteur ?

<https://innovation.iha.unistra.fr/books/robotique-educative/page/quest-ce-quun-servo-moteur>

---

## Explication du câblage complet

Maintenant que nous avons vu chaque composant individuellement, voici un résumé du câblage complet :

1. **Capteur à ultrasons** (HC-SR04) :
    - **VCC** → 5V de l'Arduino
    - **GND** → GND de l'Arduino
    - **TRIG** → D9 de l'Arduino
    - **ECHO** → D10 de l'Arduino
  2. **Servomoteur** :
    - **VCC** (fil rouge) → 5V de l'Arduino
    - **GND** (fil noir) → GND de l'Arduino
    - **Signal** (fil jaune) → D12 de l'Arduino
- 

## Conclusion et exercices pratiques

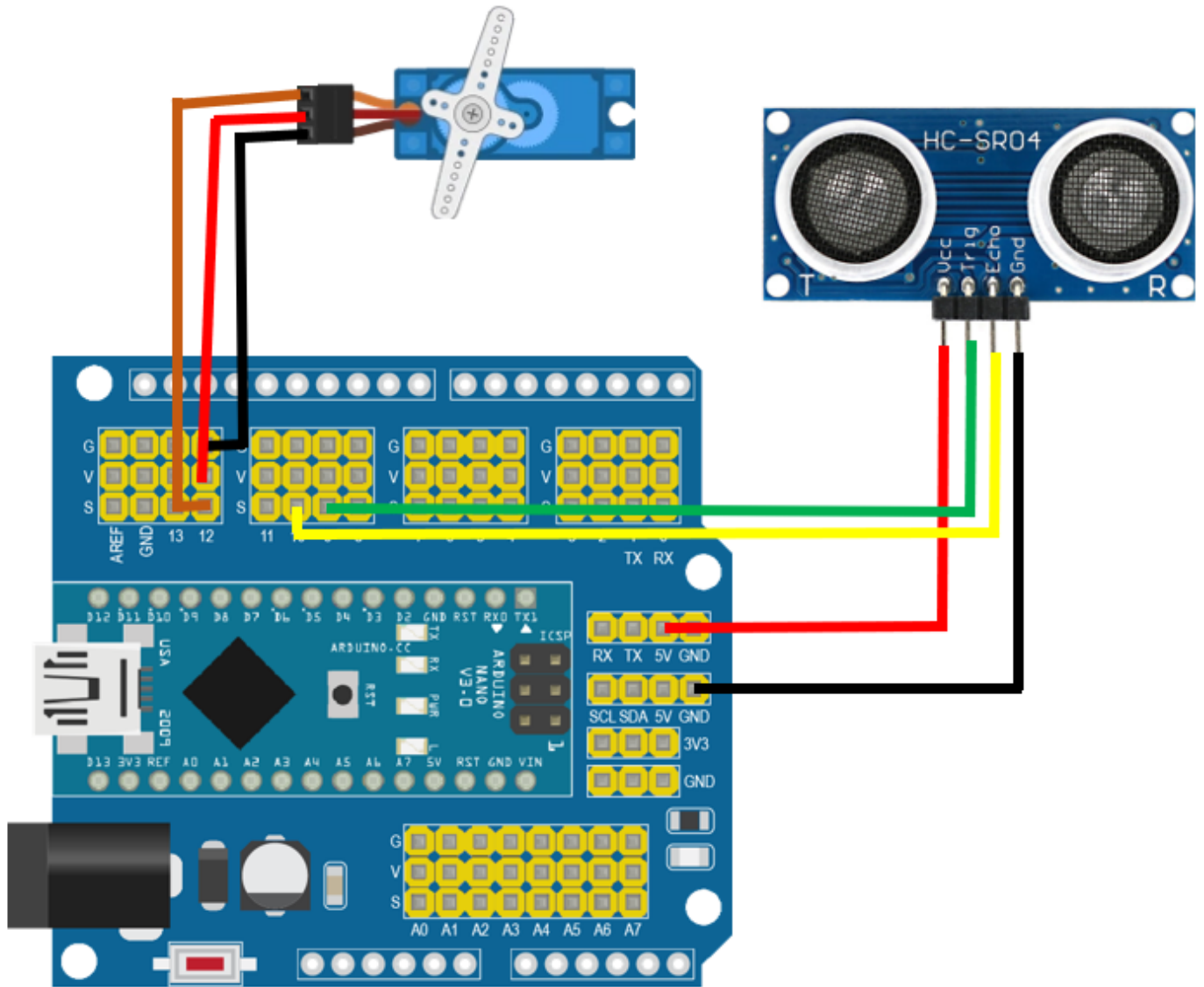
### Synthèse :

- Nous avons vu comment l'Arduino Nano fonctionne avec un shield pour faciliter les connexions.
- Nous avons compris comment le capteur à ultrasons mesure des distances et comment nous allons utiliser ces informations pour contrôler un servomoteur.
- Le servomoteur répond à des signaux PWM pour se déplacer selon des angles précis.

## Exercice pratique :

Pour terminer la séance, les participants peuvent reproduire le câblage décrit ci-dessus sur leur propre montage.

## Correction de L'exercice :



**Code à utiliser :** Le code sera introduit dans la prochaine séance, mais vous pouvez donner un aperçu du fonctionnement et les amener à réfléchir sur la façon dont les composants vont interagir ensemble.

# Phase 2 - Utilisation des composants et programmation de la poubelle connectée

## Phase 2 - Utilisation des composants et programmation de la poubelle connectée :

### Objectifs pédagogiques :

- Apprendre à utiliser le capteur à ultrasons (HC-SR04) pour mesurer des distances.
- Apprendre à contrôler le servomoteur en fonction des signaux PWM.
- Assembler les deux composants pour que le servomoteur se déplace lorsque la distance mesurée est inférieure à 5 cm.
- Comprendre l'interaction entre les capteurs et les actionneurs dans un projet Arduino.

### Matériel nécessaire :

- 1 x **Arduino Nano**
  - 1 x **Shield pour Arduino Nano** ou une breadboard
  - 1 x **Capteur à ultrasons HC-SR04**
  - 1 x **Servomoteur SG90**
  - Fils de connexion
  - 1 x **Câble USB** pour programmer l'Arduino
  - 1 x **Ordinateur** avec l'IDE Arduino installé
- 

## 1. Utilisation du capteur à ultrasons HC-SR04

### Objectif :

Les élèves vont apprendre à écrire un programme qui utilise le capteur à ultrasons pour mesurer la distance.

## Instructions :

### 1. Câblage :

- Suivez le câblage décrit dans la première séance :
  - **VCC** → 5V de l'Arduino
  - **GND** → GND de l'Arduino
  - **TRIG** → D9 de l'Arduino
  - **ECHO** → D10 de l'Arduino

### 2. Écriture du code : Demandez aux élèves d'écrire ou de copier le code suivant dans l'IDE Arduino :

```
// Déclaration des pins du capteur à ultrasons
const int trigPin = 9;
const int echoPin = 10;

void setup() {
  Serial.begin(9600); // Initialiser la communication série
  pinMode(trigPin, OUTPUT); // Définir le trigPin comme une sortie
  pinMode(echoPin, INPUT); // Définir l'echoPin comme une entrée
}

void loop() {
  // Envoi d'un signal
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Lecture de la durée du signal de retour
  long duration = pulseIn(echoPin, HIGH);
  int distance = duration * 0.034 / 2; // Calcul de la distance

  // Afficher la distance dans le moniteur série
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
}
```

```
delay(500); // Attendre un peu avant la prochaine mesure  
}
```

### 3. Tester le code :

- Demandez aux élèves de télécharger le code sur leur Arduino.
- Ouvrir le **moniteur série** (dans l'IDE Arduino) pour voir les distances mesurées.

## 2. Utilisation du servomoteur SG90

### Objectif :

Les élèves vont apprendre à contrôler le servomoteur en utilisant des signaux PWM.

### Instructions :

#### 1. Câblage :

- Câblez le servomoteur selon le schéma précédent :
  - **Rouge** (VCC) → 5V de l'Arduino
  - **Noir** (GND) → GND de l'Arduino
  - **Jaune** (Signal) → D6 de l'Arduino

#### 2. Écriture du code : Demandez aux élèves d'écrire ou de copier le code suivant dans l'IDE Arduino :

```
#include <Servo.h> // Inclure la bibliothèque Servo  
  
Servo myServo; // Créer un objet Servo  
  
void setup() {  
    myServo.attach(6); // Attacher le servomoteur à la pin D6  
    myServo.write(0); // Position initiale à 0 degrés  
}  
  
void loop() {  
    // Faire tourner le servomoteur à 90 degrés  
    myServo.write(90);  
    delay(1000); // Attendre 1 seconde  
    myServo.write(0); // Retourner à 0 degrés  
    delay(1000); // Attendre 1 seconde  
}
```

### 3. Tester le code :

- Demandez aux élèves de télécharger le code sur leur Arduino.
  - Observez le mouvement du servomoteur, qui devrait osciller entre 0 et 90 degrés.
- 

## 3. Assemblage du capteur à ultrasons et du servomoteur

### Objectif :

Les élèves vont assembler les deux composants pour que le servomoteur se déplace lorsque la distance mesurée est inférieure à 5 cm.

### Câblage :

Utiliser les connexions précédemment définies. Les deux composants doivent être câblés en parallèle à l'Arduino. Voici un résumé :

- **Capteur à ultrasons :**
  - **VCC** → 5V de l'Arduino
  - **GND** → GND de l'Arduino
  - **TRIG** → D9 de l'Arduino
  - **ECHO** → D10 de l'Arduino
- **Servomoteur :**
  - **VCC** (fil rouge) → 5V de l'Arduino
  - **GND** (fil noir) → GND de l'Arduino
  - **Signal** (fil jaune) → D6 de l'Arduino

### Écriture du code final :

Demandez aux élèves d'écrire ou de copier le code suivant, qui intègre le capteur et le servomoteur :

```
#include <Servo.h> // Inclure la bibliothèque Servo

// Déclaration des pins
const int trigPin = 9;
const int echoPin = 10;
Servo myServo;

void setup() {
  Serial.begin(9600); // Initialiser la communication série
```



```

pinMode(trigPin, OUTPUT); // Définir le trigPin comme sortie
pinMode(echoPin, INPUT);  // Définir l'echoPin comme entrée
myServo.attach(6); // Attacher le servomoteur à la pin D6
myServo.write(0); // Position initiale à 0 degrés
}

void loop() {
  // Envoyer un signal
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Lire la durée du signal de retour
  long duration = pulseIn(echoPin, HIGH);
  int distance = duration * 0.034 / 2; // Calculer la distance

  // Afficher la distance dans le moniteur série
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  // Si la distance est inférieure à 5 cm, déplacer le servomoteur
  if (distance < 5) {
    myServo.write(140); // Positionner le servomoteur à 180 degrés
    delay(1000); // Attendre 1 seconde
    myServo.write(0); // Retourner à 0 degrés
  }

  delay(500); // Attendre un peu avant la prochaine mesure
}

```

## 4. Tester le projet

### Instructions :

1. Demandez aux élèves de télécharger le code final sur leur Arduino.
2. Placez un objet à moins de 5 cm du capteur à ultrasons.

3. Observez le servomoteur se déplacer à 180 degrés pendant 1 seconde, puis revenir à sa position initiale.
- 

## 5. Conclusion et discussion

### Synthèse :

- Les élèves ont appris à utiliser le capteur à ultrasons pour mesurer des distances.
- Ils ont contrôlé un servomoteur en fonction des signaux PWM.
- Ils ont assemblé les deux composants pour créer un projet fonctionnel où le servomoteur se déplace en réponse à la détection d'un objet à moins de 5 cm.

### Questions de réflexion :

- Comment pourrait-on améliorer ce projet ? (ex. : ajouter une LED qui s'allume lorsque l'objet est détecté)
- Quelles autres applications pourraient utiliser un capteur à ultrasons et un servomoteur ensemble ?

## Projet similaire

[https://www.hackster.io/FANUEL\\_CONRAD/automatic-soap-dispenser-75abd6](https://www.hackster.io/FANUEL_CONRAD/automatic-soap-dispenser-75abd6)