

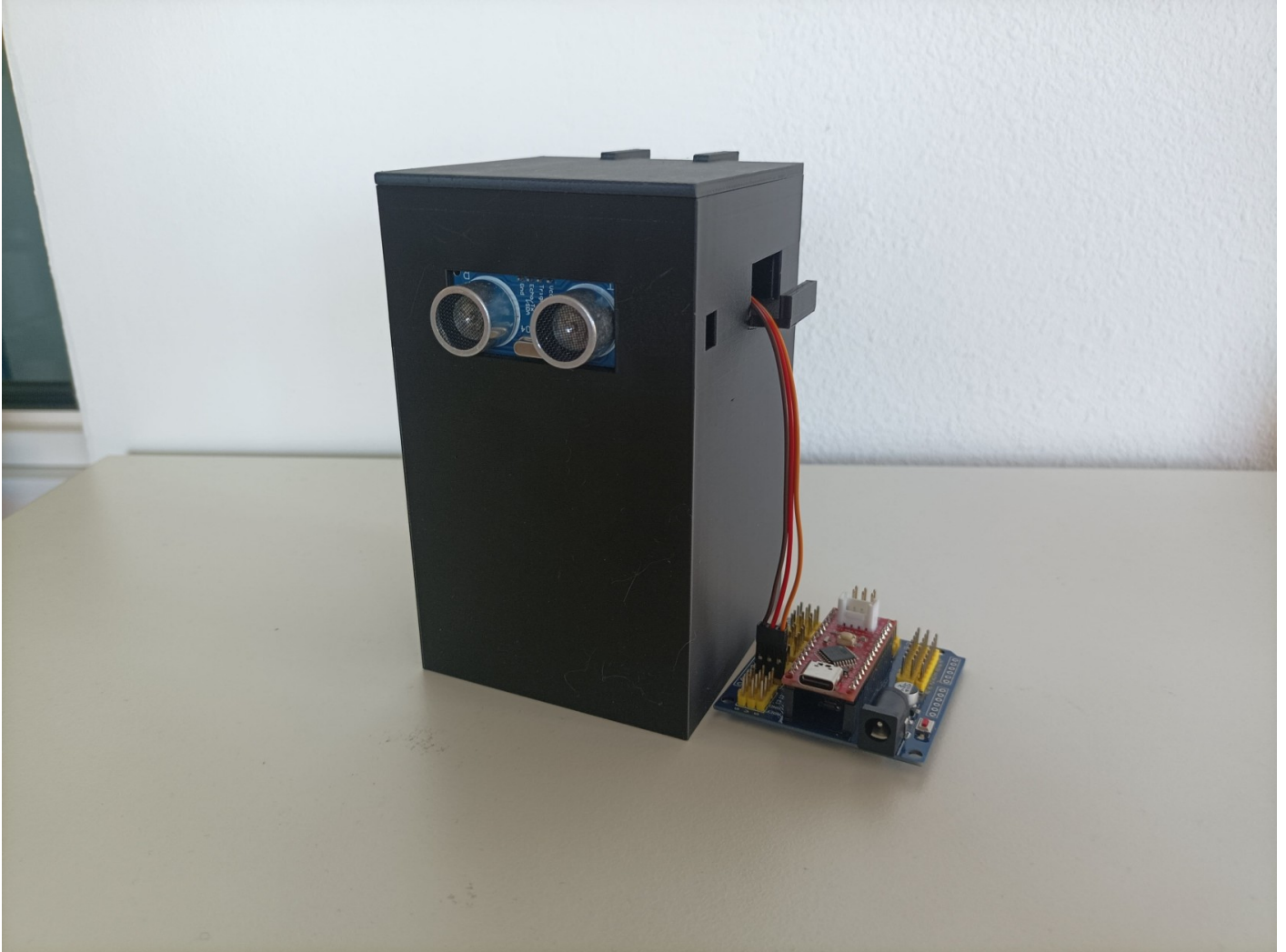
Club Robotique - Poubelle à couvercle automatique

- [Description du projet](#)
- [Modélisation en 3D du couvercle](#)
- [Fonctionnement et programmation des composants](#)
- [Assemblage et programmation de la poubelle à couvercle automatique](#)

Description du projet

Introduction

Pour ce cycle, nous allons réaliser une poubelle avec une ouverture de couvercle automatique.



Description des compétences visées :

Compétences techniques et mécaniques :

- Conception mécanique : savoir imaginer et concevoir le mécanisme d'ouverture automatique du couvercle
- Assemblage de composants : apprendre à manipuler et assembler les différentes parties (couvercle, axe, moteur, composants électroniques, etc.)
- Choix des matériaux : identifier les matériaux adaptés au projet

Compétences en électronique :

- Découverte et câblage de capteurs pour détecter une présence.
- Découverte et câblage de servomoteur pour l'ouverture/fermeture du couvercle.
- Savoir réaliser un schéma de câblage et connecter les composants sur une carte Arduino.

Compétences en programmation :

- Programmation d'un microcontrôleur Arduino : écrire un code qui gère les capteurs, les délais, et l'action du moteur.
- Logique conditionnelle : comprendre comment programmer des conditions simples (si détection → ouvrir couvercle).
- Optimisation du fonctionnement : ajuster les temps de réponse, la durée d'ouverture, etc.

Compétences transversales :

- Gestion de projet : planifier les étapes de réalisation, répartir les tâches, respecter les délais.
- Résolution de problèmes : savoir réagir face aux imprévus techniques ou fonctionnels.
- Travail en équipe.

Description du programme des séances :

~6 séances :

- ~Introduction du projet (brainstorming)
- ~2 séances sur TinkerCAD et sur l'impression 3D
- ~1 séance Actionneur : servomoteur RC 180°(déplacement)
- ~1 séance Capteur : ultrason (distance)
- ~1 séance Assemblage et programmation

Matériel :

- 1 x Arduino Nano ou compatible ([seeeduno](#), funduino,...) : ~10€
 - 1 x câble USB C : ~1€
 - 1 x [Sensor Shield](#) pour Arduino Nano : ~3€
 - 1 x servomoteur à rotation 180° ([Servomoteur 180° SG90 9g](#)) : ~2€
 - 1 x Module ultrason ([HC-SR04](#)) : ~2€
 - 1 x porte pile 4xaa : ~2€
 - Une imprimante 3D pour imprimer les pièces du projet (fichier disponible à la prochaine page)(~5€ de filament) :
- Total de 25€ par poubelle

Modélisation en 3D du couvercle

Avant de commencer à modéliser en 3D notre pièce, veuillez consulter les pages suivantes :

- Cette [page](#) expliquant comment créer une classe sur Tinkercad (utile pour le travail en groupe ou en classe).
- Cette [page](#) expliquant comment utiliser Tinkercad pour modifier un fichier 3D (Une séance pour apprendre à utiliser tinkercad avant de modéliser la pièce finale peut-être envisagé)

Commençons à modéliser en 3D le couvercle de la poubelle à couvercle automatique.



Préparation de l'environnement

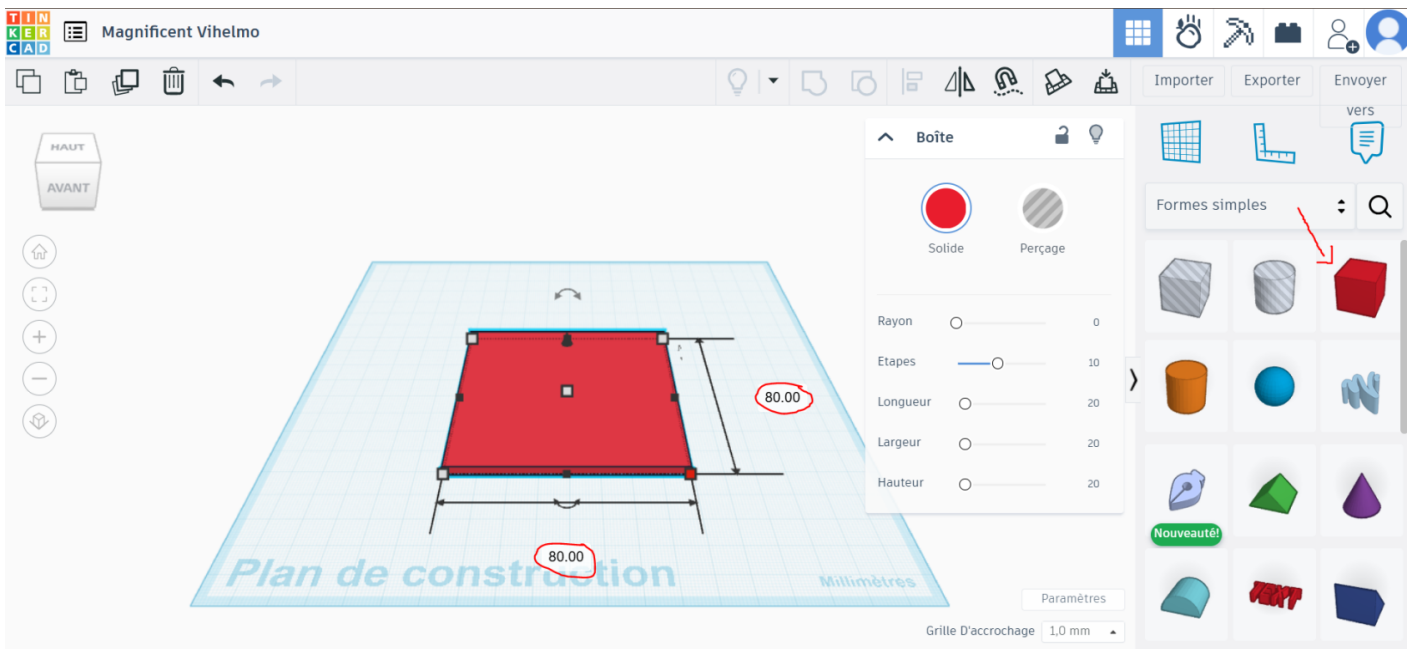
Créer une classe sur Tinkercad et faire rejoindre tous les élèves.

Comment créer une classe sur tinkercad :

<https://innovation.iha.unistra.fr/books/2-fablab-formation-machines-logiciels/page/creation-dune-classe-tinkercad>

Réaliser un pavé droit de dimension :

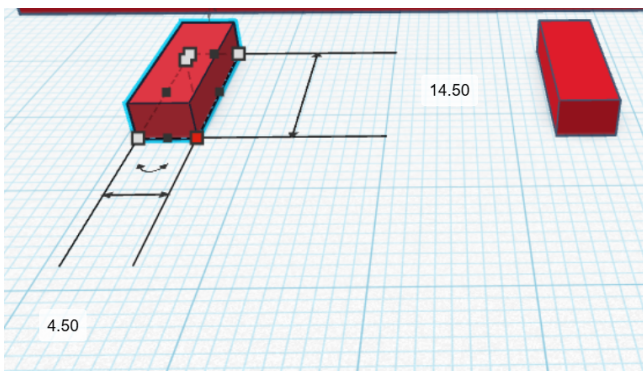
- 80 mm de Longueur
- 80 mm de largeur
- 3 mm de hauteur



Réaliser deux pavés droits de dimensions :

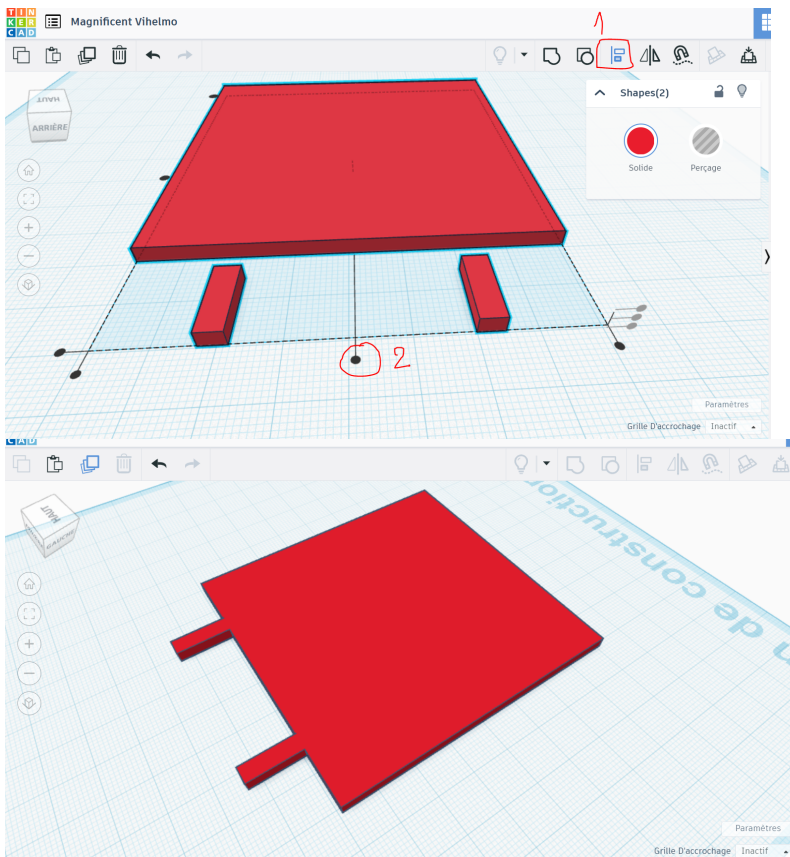
- 14.5 mm de Longueur
- 4.5 mm de largeur
- 3 mm de hauteur

Les placer avec un espacement de 43 mm entre eux, les sélectionner et les regrouper (CTRL+G).



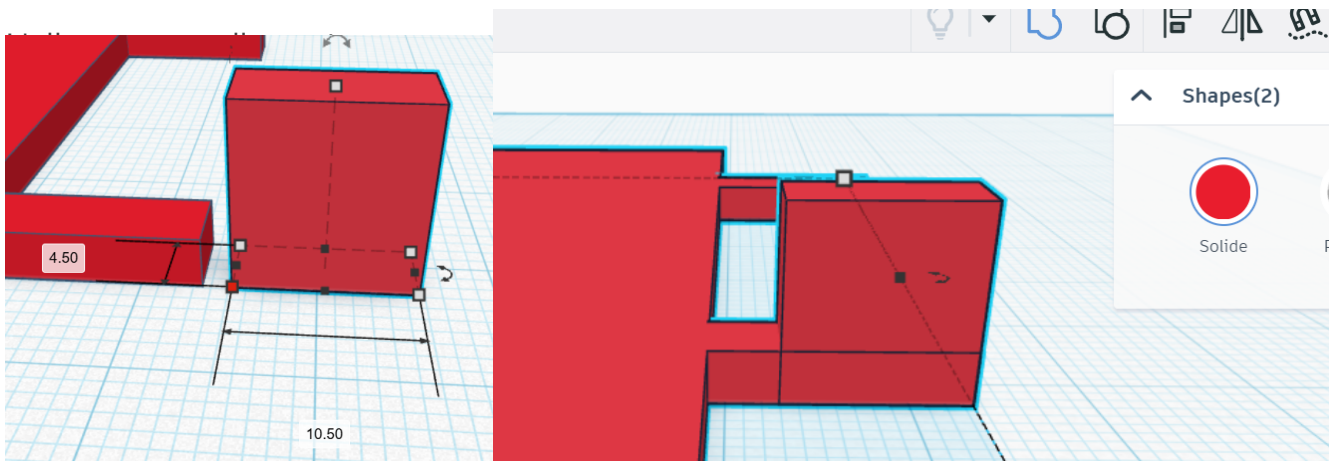
Center les deux pavés droits la base du couvercle comme l'image ci-dessous à gauche et aligner manuellement les pavés droit à la base comme l'image ci-dessous à droite et regrouper le tout (CTRL+G)

Astuce : afin d'aligner le tout plus facilement, vous pouvez rendre "Inactif" la "Grille D'accrochage" en bas à droite de l'interface de Tinkercad.

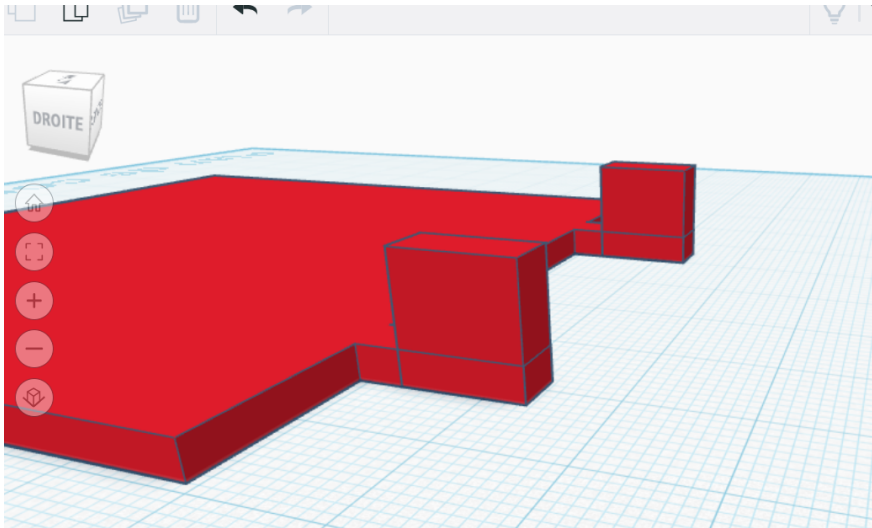


Réaliser un pavé droit de dimension :

- 10.5 mm de Longueur
- 4.5 mm de largeur
- 11 mm de hauteur



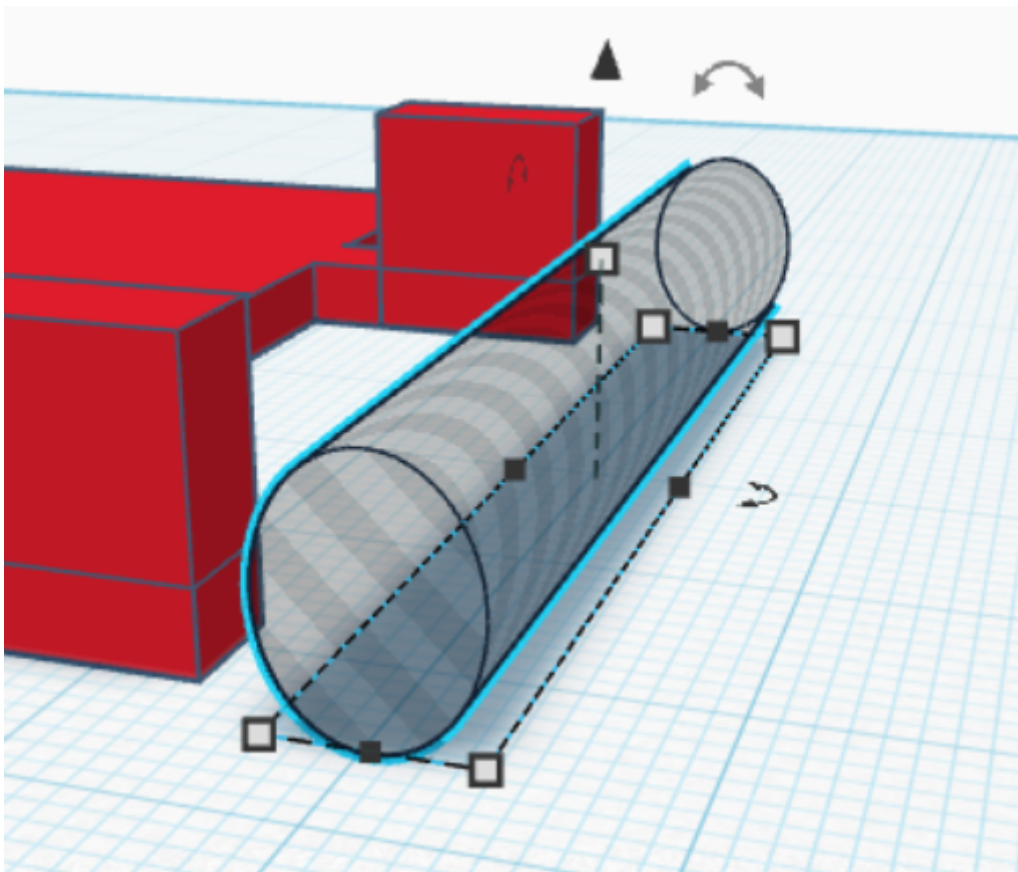
Faire de même pour l'autre coté.



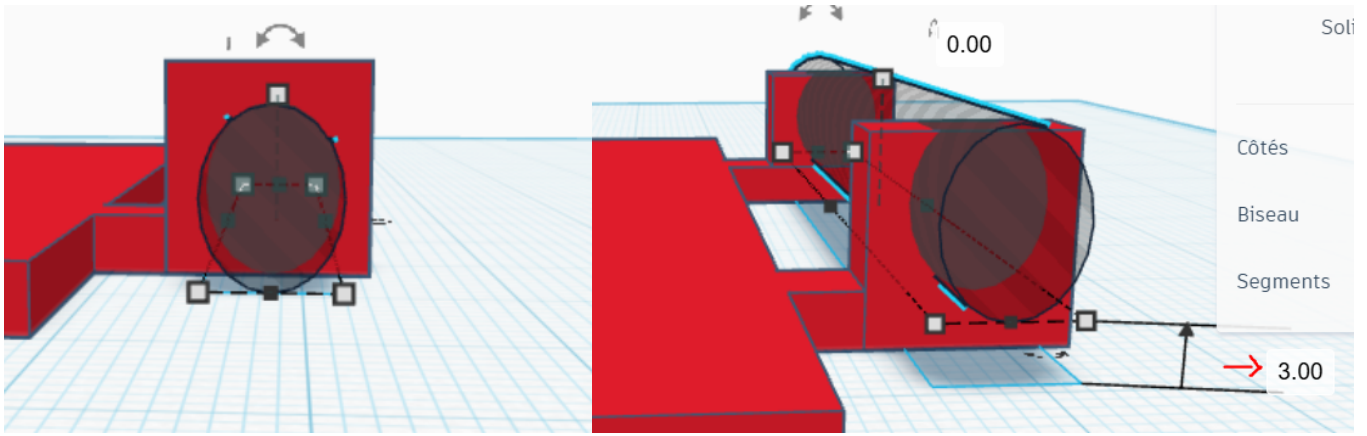
Réaliser un cylindre de dimension :

- 10.5 mm de Longueur
- 4.5 mm de largeur
- 11 mm de hauteur

Et le centrer avec l'outil aligner par rapport à la pièce réalisée précédemment.



Le centrer maintenant par rapport à cette partie de la pièce (voir photo ci-dessous à gauche) et le surélever (avec la flèche noir entourée en rouge sur la photo ci-dessous à droite) de 3 mm. A présent, sélectionner le tout et les assembler.



Fonctionnement et programmation des composants

Objectif :

Comprendre le fonctionnement des composants du projet et apprendre à les programmer

1. Introduction à l'Arduino Nano

Qu'est-ce qu'un Arduino ?

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

2. Capteur à ultrasons HC-SR04

Qu'est ce qu'un capteur à ultrason et comment il fonctionne ?

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

3. Servomoteur SG90 180°

Qu'est ce qu'un servomoteur et comment il fonctionne ?

<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

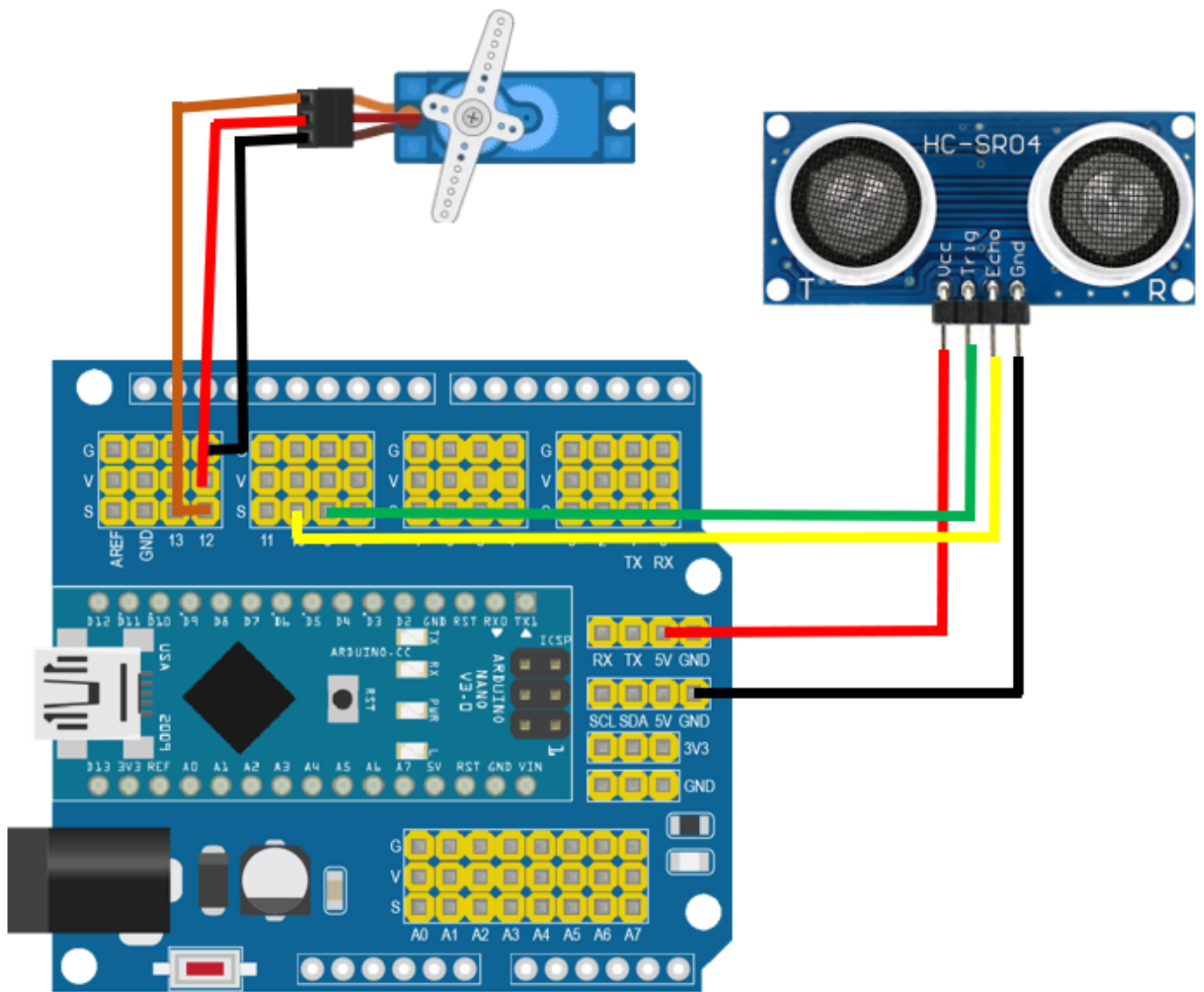
Synthèse :

- 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.
- Nous avons compris comment fonctionne le servomoteur et comment le contrôler.

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.

Assemblage et programmation de la poubelle à couvercle automatique

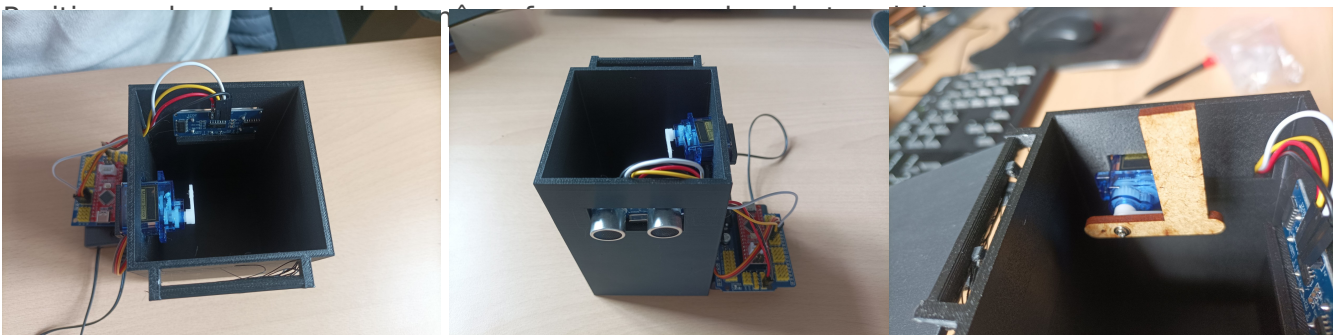
1. Utilisation du capteur à ultrasons HC-SR04

Objectif :

Gestion et programmation des capteurs/actionneurs de la poubelle à couvercle automatique dans son ensemble.

Assemblage :

Le modèle 3D de la poubelle à imprimer en 3D se trouve en pièce jointe du wiki



Câblage :

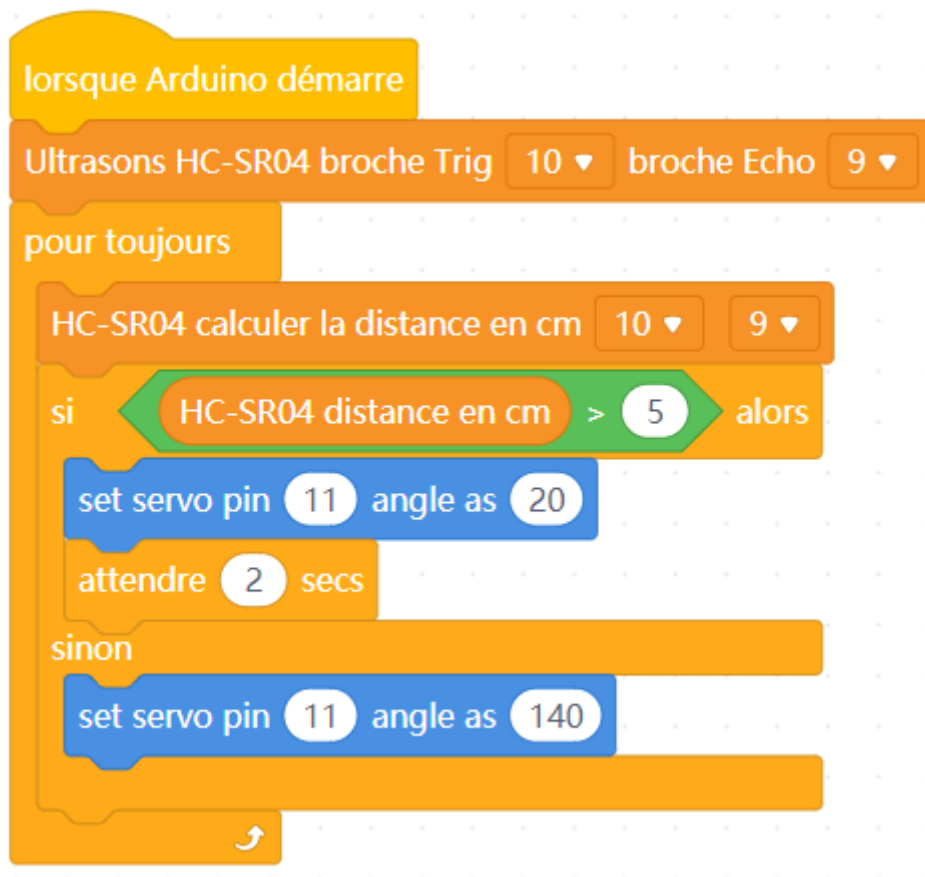
Vous référez à la page précédente . Voici un résumé :

- **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) → D11 de l'Arduino

Programmation final :

Programmation par bloc avec Mblock :

Demander aux élèves de réaliser le code final tout en les aidants si besoin. Voici le code final :



Programmation avec arduino IDE :

En fonction du niveau, demander aux élèves d'écrire ou de copier le code suivant tout en expliquant ce dernier.

```
#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
}
```

Conclusion et discussion

Synthèse des compétences acquises :

Compétences techniques et mécaniques :

- Conception mécanique : savoir imaginer et concevoir le mécanisme d'ouverture automatique du couvercle
- Assemblage de composants : apprendre à manipuler et assembler les différentes parties (couvercle, axe, moteur, composants électroniques, etc.)
- Choix des matériaux : identifier les matériaux adaptés au projet

Compétences en électronique :

- Découverte et câblage de capteurs pour détecter une présence.
- Découverte et câblage de servomoteur pour l'ouverture/fermeture du couvercle.
- Savoir réaliser un schéma de câblage et connecter les composants sur une carte Arduino.

Compétences en programmation :

- Programmation d'un microcontrôleur Arduino : écrire un code qui gère les capteurs, les délais, et l'action du moteur.
- Logique conditionnelle : comprendre comment programmer des conditions simples (si détection → ouvrir couvercle).
- Optimisation du fonctionnement : ajuster les temps de réponse, la durée d'ouverture, etc.

Compétences transversales :

- Gestion de projet : planifier les étapes de réalisation, répartir les tâches, respecter les délais.
- Résolution de problèmes : savoir réagir face aux imprévus techniques ou fonctionnels.
- Travail en équipe.

Perspectives :

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

- etc...

Projet similaire

https://www.hackster.io/FANUEL_CONRAD/automatic-soap-dispenser-75abd6