

Projet final : Bateau sur l'eau

- Yasmine, Rita, Maélanne, Marya

Objectif :

Notre projet final consiste à concevoir un prototype d'un bateau intégrant un système Arduino et une hélice. Ce compte rendu a pour objectif de documenter l'ensemble du processus, intégrant les technologies vues en atelier dont: conception et impression 3D, conception 2D et découpe laser, ou électronique numérique.

Deux approches sont possibles : créer un prototype visuellement proche de l'objet final ou développer un système électronique simulant son comportement — voire une combinaison des deux.

PARTIE 1 : Contexte et Objectifs du Projet : Bateau Décoratif Animé avec Système Arduino

1. Définition du projet

Ce projet ne se contente pas d'être un simple objet décoratif statique. Imaginez un bureau ou une étagère où un bateau miniature prend vie, son hélice tournant au gré des réglages, offrant un spectacle à la fois hypnotique et technologique. Inspiré par l'univers de LEGO Technic, ce projet cherche à allier esthétique, innovation et interactivité.

Dans un monde où la décoration devient de plus en plus intelligente et modulable, l'idée est d'introduire un objet interactif et éducatif, accessible aussi bien aux amateurs de design qu'aux passionnés de technologie. Pourquoi se limiter à des objets décoratifs statiques quand on peut leur

donner du **mouvement** et une **dimension ludique** ?

2. Public cible

Le projet vise plusieurs types d'utilisateurs :

- Les **passionnés de LEGO Technic et de modélisme**, qui cherchent des moyens d'intégrer de nouveaux mécanismes motorisés à leurs créations.
- Les **amateurs de design d'intérieur** qui veulent apporter une touche d'originalité à leur espace avec un objet décoratif interactif.
- Les **adeptes du DIY** et de **l'électronique**, qui aiment construire, programmer et expérimenter avec des circuits Arduino.

En offrant une expérience immersive et modulaire, ce projet s'adresse aussi bien aux curieux souhaitant apprendre l'électronique qu'aux passionnés de modélisme cherchant un défi créatif.

3. Fonctionnalités

Principale : Animation de l'hélice via un potentiomètre

L'utilisateur peut ajuster la vitesse de rotation en tournant un bouton, ce qui crée une sensation de contrôle fluide et intuitif, comme s'il pilotait réellement un bateau en miniature.

Secondaires :

- Un design soigné et personnalisable, qui permet d'intégrer le bateau dans différents styles d'environnement (bureau, salon, vitrine).
- Une modularité : possibilité d'ajouter d'autres animations, d'améliorer l'éclairage ou d'intégrer des capteurs supplémentaires.
- Compatibilité avec les tendances LEGO Technic et modélisme, facilitant son intégration dans des collections existantes.

4. Analyse de la problématique et veille sur l'existant

Pourquoi ce projet est-il pertinent aujourd'hui ?

L'univers du modélisme et de l'électronique interactive connaît un essor considérable. **Cependant, la plupart des objets décoratifs interactifs sur le marché sont soit trop complexes, soit limités à des fonctionnalités prédéfinies.**

Étude de l'existant :

- Modèles de bateaux animés : souvent destinés au modélisme naval, nécessitent un montage complexe et un espace dédié.
- LEGO Technic motorisés : proposent des mécanismes intéressants mais ne sont pas spécifiquement pensés comme des objets décoratifs.

- Objets décoratifs interactifs : principalement lumineux ou statiques, avec peu d'éléments en mouvement mécanique.
 - Ce projet cherche donc à combler ce vide en proposant un objet à la fois fonctionnel, décoratif et accessible.
-

5. Lean Canvas

Problème :

Le marché actuel des objets décoratifs interactifs présente deux limitations majeures :

1. Les produits existants sont souvent trop techniques et difficiles à assembler pour un public non expert (ex. : modélisme naval traditionnel).
2. À l'inverse, les objets interactifs accessibles manquent de modularité ou d'intérêt technologique (ex. : simples lampes LED décoratives).

Il existe un manque d'alternatives **ludiques, éducatives et personnalisables** qui allient à la fois esthétique, technologie et simplicité de prise en main.

Solution :

Développement d'un **bateau miniature interactif motorisé**, intégrant un système Arduino :

- **Contrôle de l'hélice via un potentiomètre**, pour simuler une navigation.
- **Design personnalisable** (formes, couleurs, finitions), adaptable à différents environnements (bureau, salon, vitrine, etc.).
- **Structure modulaire** : possibilité d'ajouter capteurs, éclairages LED ou autres animations.
- **Facilité d'assemblage** grâce à l'impression 3D et à la documentation fournie.

Segments de clientèle :

- **Modélistes** recherchant un projet créatif mêlant mécanique et esthétique.
 - **Passionnés de LEGO Technic** curieux d'élargir leur univers avec des composants Arduino.
 - **Makers et bricoleurs DIY** attirés par les projets accessibles et évolutifs.
 - **Amateurs de décoration geek/design tech** souhaitant intégrer un objet original à leur intérieur.
 - **Enseignants ou animateurs STEM** à la recherche de supports pédagogiques interactifs.
-

Proposition de valeur unique :

Un objet **esthétique et intelligent**, à la **frontière entre design, technologie et jeu** :

" Un mini-bateau décoratif, programmable et personnalisable, qui donne vie à votre espace tout en stimulant votre créativité."

Canaux de distribution :

- **Plateformes de partage et de vente de projets DIY** : Thingiverse (modèles 3D gratuits), Etsy (kits ou objets finis), Cults3D.
 - **Boutiques spécialisées** en électronique ou modélisme (en ligne ou physiques).
 - **Communautés de passionnés** : forums Arduino, groupes Facebook, Reddit, Instructables.
 - **Événements et salons** : Maker Faires, salons de modélisme ou de l'électronique.
 - **Réseaux pédagogiques** : établissements scolaires, Fablabs, associations éducatives
 - **Structure de coûts** : Matériaux, composants électroniques, outils de fabrication.
 - **Sources de revenus** : Vente de kits, fichiers de conception 3D, tutoriels payants.
-

4. Choix techniques

Arduino et électronique :

Nous avons utilisé une carte Arduino pour piloter le moteur à courant continu, avec un potentiomètre permettant de faire varier sa vitesse. L'électronique comprend également une diode pour la protection contre les retours de courant, une résistance, et un transistor servant d'interrupteur électronique entre la carte et le moteur.

Matériaux de construction :

La coque du bateau et ses composants ont été imprimés en 3D à l'aide de plastique PLA, un matériau à la fois léger et robuste. Le pont a été conçu à partir d'une planche de bois de 3 mm d'épaisseur, gravée au laser pour un rendu esthétique (effet parquet) et un bon maintien mécanique.

Justification :

Ces choix ont été guidés par plusieurs critères :

- **Compatibilité** : L'Arduino est facile à intégrer avec d'autres composants électroniques standards.
- **Accessibilité** : Tous les matériaux sont facilement disponibles en fablab.
- **Esthétique** : Le bois apporte un rendu visuel chaleureux et réaliste.
- **Robustesse** : L'ensemble offre une bonne durabilité pour un prototype manipulable.

MVP (Minimum Viable Product) :

Le premier objectif était d'obtenir un prototype fonctionnel capable d'animer une hélice grâce à un moteur contrôlé par un potentiomètre. Ce MVP a été atteint après une série de tests.

Planification :

Le projet a été mené en plusieurs étapes :

1. **Conception** (croquis, modélisation 3D, choix des composants),
2. **Fabrication** (impression 3D, découpe laser, montage électronique),

3. **Tests** (fonctionnement de l'hélice, ajustements du pont et de la coque),
4. **Améliorations** (révision des dimensions, stabilisation du moteur).

Méthodologie :

Nous avons adopté une approche itérative : à chaque étape, des tests ont été réalisés, suivis d'ajustements. Cela nous a permis de progresser rapidement tout en corrigeant les erreurs au fur et à mesure.

5. Gestion de projet

- **MVP (Minimum Viable Product) :**

Le premier objectif était d'obtenir un prototype fonctionnel capable d'animer une hélice grâce à un moteur contrôlé par un potentiomètre. Ce MVP a été atteint après une série de tests.

Planification :

Le projet a été mené en plusieurs étapes :

1. **Conception** (croquis, modélisation 3D, choix des composants),
2. **Fabrication** (impression 3D, découpe laser, montage électronique),
3. **Tests** (fonctionnement de l'hélice, ajustements du pont et de la coque),
4. **Améliorations** (révision des dimensions, stabilisation du moteur).

Méthodologie :

Nous avons adopté une approche itérative : à chaque étape, des tests ont été réalisés, suivis d'ajustements. Cela nous a permis de progresser rapidement tout en corrigeant les erreurs au fur et à mesure.

6. Conception et fabrication

- **Croquis et dimensions** : Le bateau a été pensé à partir de formes simples sur Tinkercad (carrés, paraboloïde), en adaptant les proportions selon les composants à intégrer.
 - **Liste des matériaux et composants** :
 - PLA pour l'impression 3D,
 - Bois 3 mm pour le pont,
 - Moteur 5V CC,
 - Potentiomètre,
 - Diode, résistance, transistor,
 - Arduino Uno.
 - **Fichiers 3D et captures d'écran** : Les formes ont été réalisées sur Tinkercad, puis ajustées pour accueillir tous les composants. Le pont a été modélisé sur Inkscape avec vectorisation d'un motif parquet.
 - **Code Arduino commenté** : *Voir la partie détaillée du code dans la section précédente. Chaque ligne est expliquée pour faciliter la compréhension et la maintenance.*
-

7. Prototypage et tests

- **Photos et analyse des tests :**

Des photos ont été prises à chaque étape de fabrication : impression de la coque, gravure du pont, intégration des composants électroniques. Les tests ont montré que le moteur répond bien au potentiomètre, avec une vitesse modulable.

- **Documentation des erreurs et ajustements :**

Des erreurs de dimension ont été constatées lors des premières impressions : certaines ouvertures étaient trop petites pour les composants. Nous avons corrigé cela en agrandissant les formes et en recalibrant les perçages.

- **Analyse du produit final :**

Le prototype fonctionne conformément à l'objectif initial. L'hélice tourne, la structure est stable, et les composants tiennent bien en place. Le rendu visuel est propre, grâce à l'effet bois et à la finition des formes.

8. Améliorations et évolutions possibles

- **Intégration d'une télécommande** (par infrarouge ou Bluetooth) pour piloter le moteur à distance.
- **Optimisation du design et des matériaux** pour rendre le bateau plus léger ou étanche.
- **Ajout d'autres mouvements animés** comme une direction (servo moteur), un mat mobile ou des LED d'ambiance.

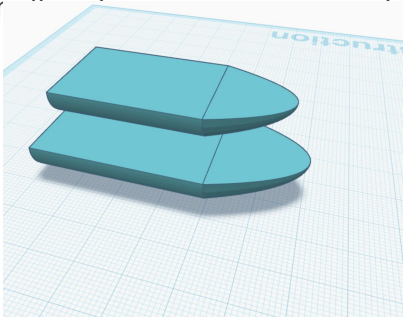
9. Ressources et inspirations

- **Tutoriels** : Documentation Arduino officielle, forums Instructables, vidéos YouTube sur les bases de la motorisation.
- **Références techniques** : Schémas de circuits Arduino de base, articles sur l'impression 3D et la découpe laser.
- **Inspirations** : Design nautique simplifié, LEGO Technic pour les systèmes mécaniques, et modélisme naval pour la structure globale.

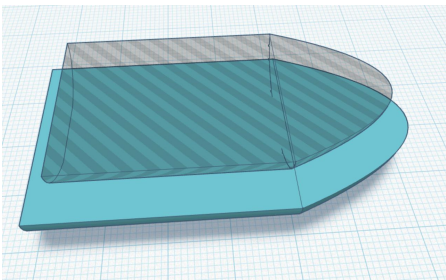
PARTIE 2 : Exécution et conception du design sur Tinkercad

A/ Création de la forme 1 : la coque du bateau

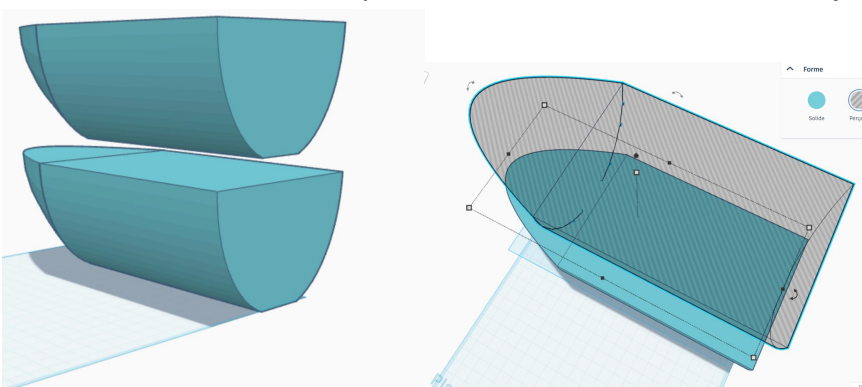
Pour débiter, nous avons créé deux carrés, puis ajouté un paraboloïde que nous avons coupé en deux. Ensuite, nous avons assemblé les différentes parties ensemble avant de dupliquer la première forme afin d'obtenir la forme intérieure du bateau. Cette étape nous a permis de déterminer les dimensions intérieures à ajuster. La deuxième forme a été réduite de deux r



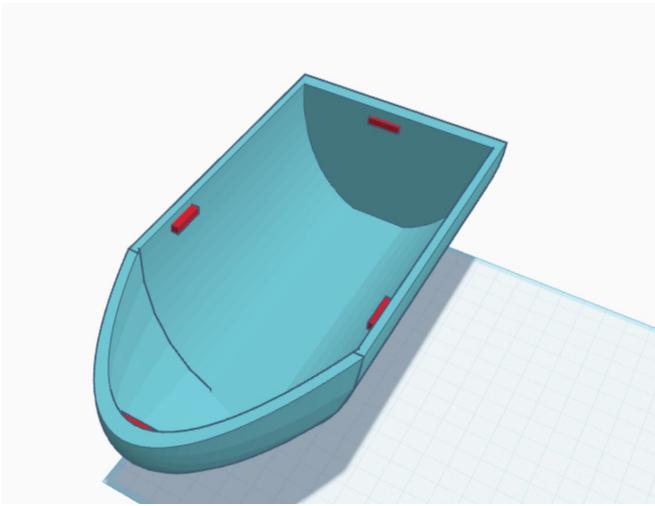
La plus petite forme a ensuite été percée pour permettre l'intégration des différents composants.



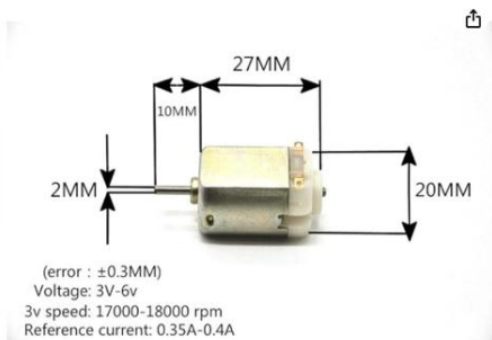
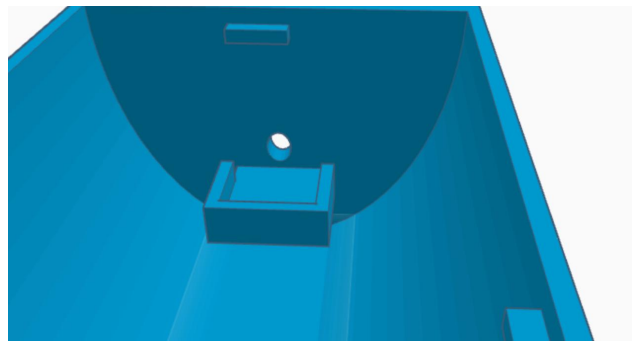
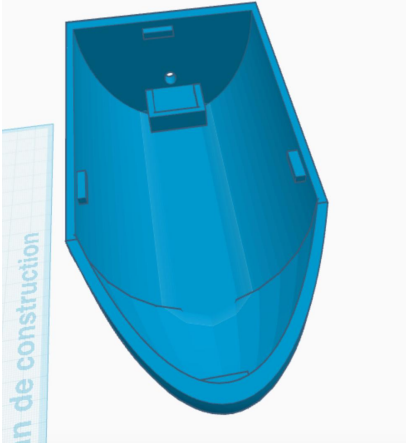
Cependant, après vérification, nous nous sommes rendus compte que les dimensions de nos formes initiales n'étaient pas correctes. Nous avons donc ajusté et agrandi les formes, puis repris



Ensuite, nous avons procédé à l'installation de cales (indiquées en rouge sur l'image ci-dessous) afin de soutenir la plateforme supérieure du bateau (le pont), qui sera réalisée par la suite.



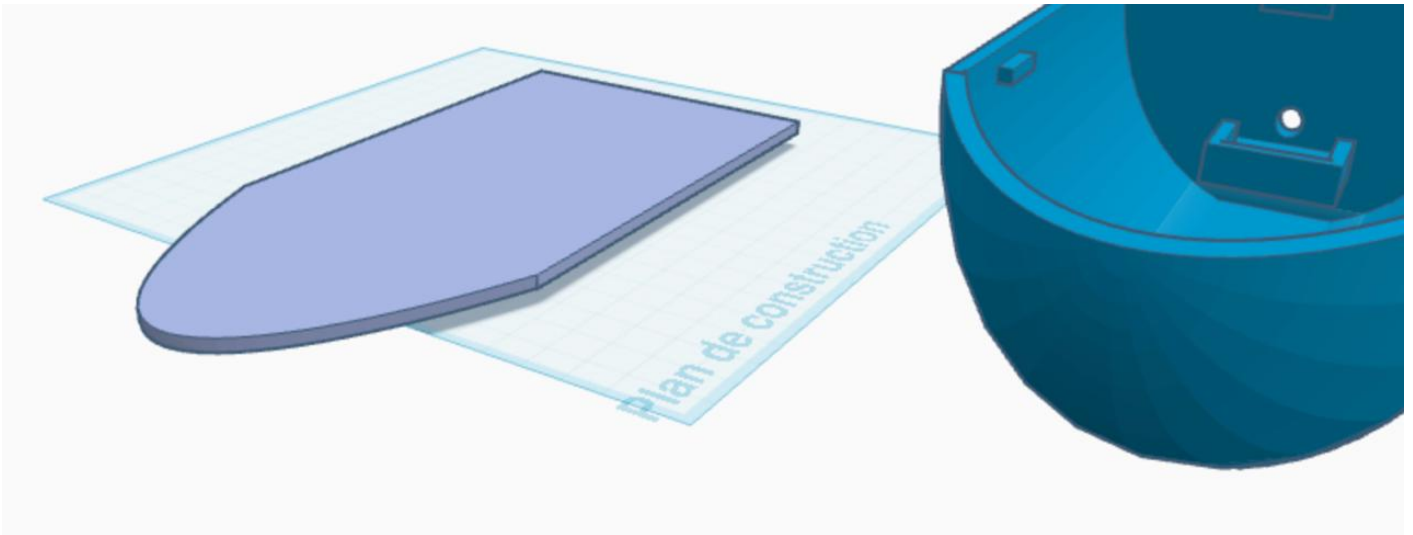
Nous avons également intégré un petit support pour le moteur à courant continu 5V (voir image ci-dessous) permettant le passage de la tige de l'hélice à l'extérieur du bateau.



B/ Création de la forme 2: le pont/plateforme du bateau

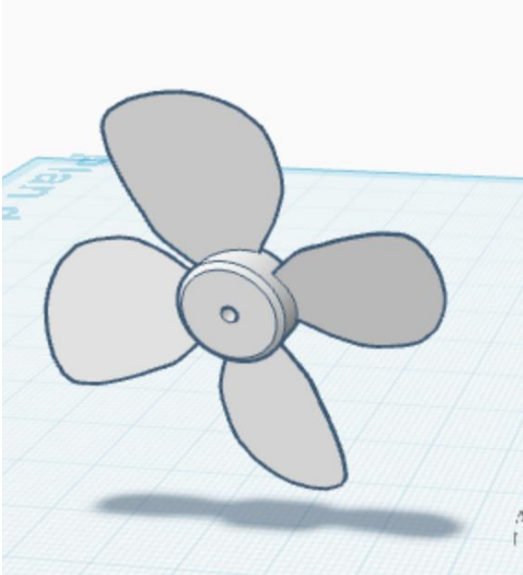
Une fois la coque de bateau terminée, nous avons modélisé le pont de manière à ce qu'il s'ajuste parfaitement à la forme de la coque, en s'appuyant sur les cales. Pour cette étape, nous avons utilisé une planche de bois de 3 mm d'épaisseur, découpée au laser, afin d'obtenir une plateforme

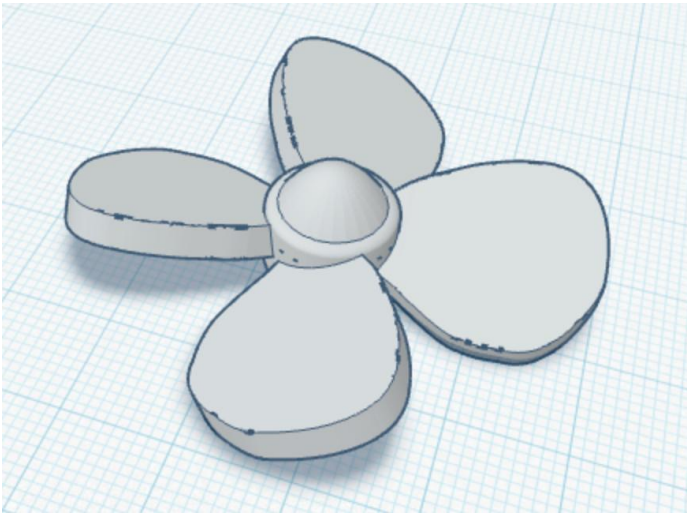
solide et précise.



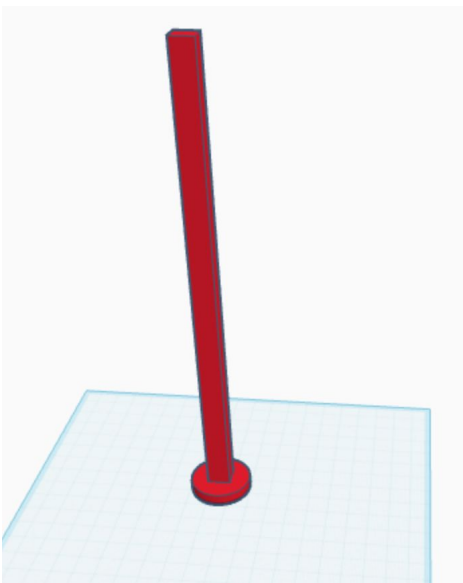
C/ Création de l'hélice et du mat

Pour concevoir notre hélice, nous nous sommes basées sur l'un des modèles proposés par la plateforme, comme celui illustré ci-dessous :





Et puis nous avons modélisé le mat comme ci-dessous :

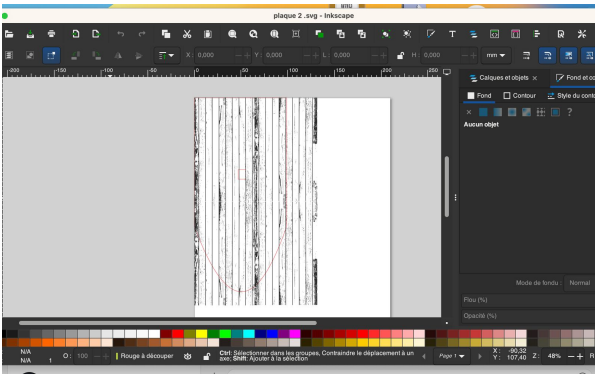
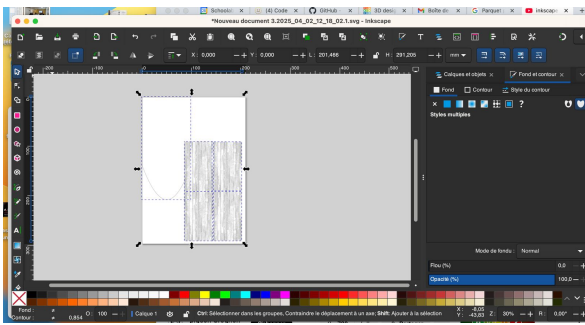
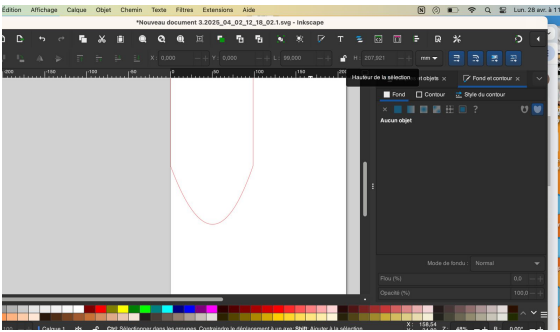


PARTIE 3 : Création du pont sur Inkscape

Pour le pont, nous avons choisi de lui donner la forme que nous souhaitons pour qu'on puisse l'insérer dans le bateau imprimé en 3D. Et, pour obtenir l'effet parquet, nous avons sélectionné un motif de parquet sur internet que nous avons passé en noir et blanc puis vectorisé afin de créer la gravure.

Ensuite, nous avons ajouté un trou pour le mât.

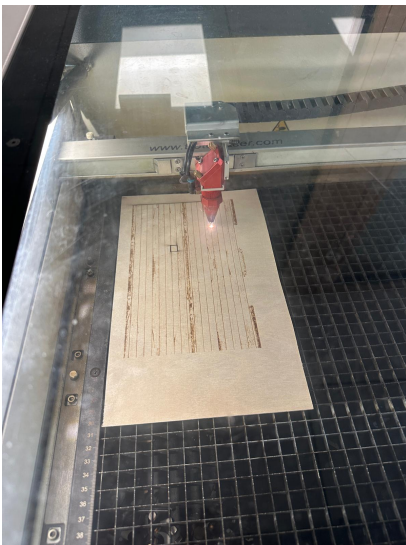
Enfin, nous avons exporté le fichier au format SVG pour l'utiliser avec la découpeuse laser.



Cependant, une fois le pont imprimé, nous avons constaté qu'il ne correspondait pas aux dimensions prévues et ne s'insérait pas correctement dans la coque (visible à gauche sur la photo ci-dessous). Nous avons donc repris les mesures afin d'adapter le pont pour qu'il repose sur l'ensemble de la coque, ce qui nous permettra de le fixer ultérieurement à l'aide de colle.



Puis nous avons réalisé une deuxième découpe laser pour obtenir notre nouveau pont.

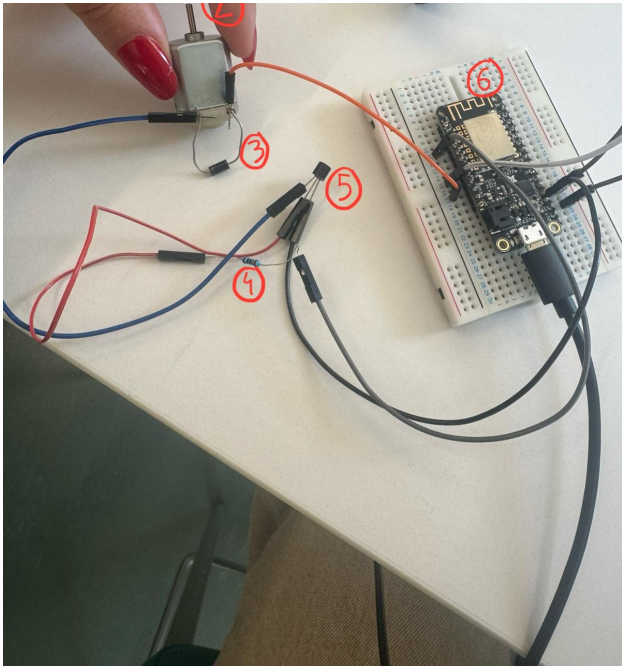




PARTIE 4 : Le montage Arduino

Pour le montage Arduino, nous avons utilisé les composants suivants (numérotés selon la légende de la photo) :

1. **Potentiomètre** - Sert à ajuster la tension en entrée et donc à contrôler la vitesse du moteur.
2. **Moteur** - Convertit l'énergie électrique en énergie mécanique (rotation).
3. **Diode** - Protège le circuit contre les retours de courant générés par le moteur lorsqu'il s'arrête.
4. **Résistance** - Limite le courant électrique pour éviter d'endommager les composants.
5. **Transistor** - Fait office d'interrupteur électronique : il permet de contrôler la mise en marche du moteur via l'Arduino.
6. **Carte Arduino** - Sert de contrôleur principal : elle envoie les signaux nécessaires pour piloter le montage.



PARTIE 5 : Le code

- **Test initial du moteur**

Nous avons tout d'abord réalisé un premier programme destiné à vérifier le bon fonctionnement du moteur.

Le test a été concluant : le moteur fonctionne correctement.

```
#define POTENTIOMETER_PIN 12 // Entrée analogique du potentiomètre
#define MOTOR_PIN 16 // Sortie PWM vers le transistor

void setup() {
  pinMode(MOTOR_PIN, OUTPUT);
}

void loop() {

  analogWrite(MOTOR_PIN, motorSpeed); // Envoyer la valeur PWM au moteur
  delay(10);
}
```

- **Ajout du potentiomètre (code final)**

Dans un second temps, nous avons modifié le code en ajoutant deux lignes supplémentaires (**code ci-dessous, ligne 9 et 10**) afin d'intégrer le potentiomètre au circuit.

Cette étape a permis de connecter et d'utiliser le potentiomètre avec succès.

```
#define POTENTIOMETER_PIN 12 // Entrée analogique du potentiomètre
#define MOTOR_PIN 16 // Sortie PWM vers le transistor

void setup() {
  pinMode(MOTOR_PIN, OUTPUT);
}

void loop() {
  int potValue = analogRead(POTENTIOMETER_PIN); // Lire la valeur du potentiomètre (0-1023)
  int motorSpeed = map(potValue, 0, 1023, 0, 255); // Convertir la valeur en PWM (0-255)

  analogWrite(MOTOR_PIN, motorSpeed); // Envoyer la valeur PWM au moteur
  delay(10);
}
```

Voici l'explication détaillée du code utilisé :

1. **#define POTENTIOMETER_PIN 12** // Entrée analogique du potentiomètre

Rôle : Cette ligne définit un nom (POTENTIOMETER_PIN) pour représenter le numéro de la broche 12.

Pourquoi ce choix :

#define est utilisé pour rendre le code plus lisible et facile à modifier : si un jour on change la broche du potentiomètre, il suffira de changer ce numéro ici, sans chercher partout dans le code.

2. **#define MOTOR_PIN 16** // Sortie PWM vers le transistor

Rôle : De la même manière, cette ligne définit MOTOR_PIN pour la broche 16, qui pilote le moteur.

3. **void setup() {**

Rôle : Début de la fonction setup(), appelée une seule fois au démarrage du microcontrôleur.

C'est la structure standard d'un programme Arduino. Tout ce qui est initialisation doit se faire dans setup().

Cela permet d'initialiser les paramètres du système dès le lancement.

4. **pinMode(MOTOR_PIN, OUTPUT);**

Rôle : Définit la broche du moteur (MOTOR_PIN) comme une sortie.

Il est nécessaire pour pouvoir envoyer un signal PWM depuis cette broche. Sinon, par défaut, les broches sont en entrée.

5. **}**

Rôle : Fin de la fonction setup().

(Syntaxe obligatoire en C/C++ pour délimiter les blocs de fonctions)

6. **void loop() {**

Rôle : Début de la fonction loop(), appelée en boucle continue tant que la carte est alimentée.

C'est ici que le système s'adapte en temps réel aux changements du potentiomètre.

7. **int potValue = analogRead(POTENTIOMETER_PIN);**

Rôle : Lit la tension en entrée sur la broche du potentiomètre (entre 0V et 5V) et la convertit en une valeur numérique entre 0 et 1023.

Cette ligne de code permet de capter la position du potentiomètre, qui déterminera la vitesse du moteur.

8. **int motorSpeed = map(potValue, 0, 1023, 0, 255);**

Rôle : Convertit la valeur lue (0-1023) en une plage plus petite (0-255) adaptée au PWM et assure que la valeur envoyée au moteur est proportionnelle à la lecture du potentiomètre.

9. **analogWrite(MOTOR_PIN, motorSpeed);**

Rôle : Génère un signal PWM correspondant à motorSpeed sur la broche du moteur, elle contrôle effectivement la vitesse du moteur en fonction de la position du potentiomètre.

10. **delay(10);**

Rôle : Fait une courte pause de 10 millisecondes avant de recommencer la boucle.

Sans delay, la boucle loop() tournerait trop vite, ce qui pourrait provoquer des lectures instables ou inutiles.

10 ms est un bon compromis : assez rapide pour être réactif, mais pas inutilement rapide.

Ainsi, on assure une stabilité de la lecture et on réduit la charge inutile du microcontrôleur.

11. **}**

Rôle : Fin de la fonction loop().

(Syntaxe obligatoire pour clôturer un bloc de fonction)

Revision #9

Created 31 March 2025 12:26:52 by El Noueiry Marya

Updated 24 September 2025 09:23:22 by Hamed Yasmine