



Système de traduction et d'apprentissage du Braille

Présentation d'un tutoriel notre projet de classe de terminale STI2D afin de le reproduire chez soi

 Difficulté **Difficile**

 Durée **72 heure(s)**

 Catégories **Électronique, Bien-être & Santé**

 Coût **250 EUR (€)**

Sommaire

- Étape 1 - Découpe des pièces en bois du boîtier
- Étape 2 - Assemblage des pièces en bois
- Étape 3 - Configuration de la carte électronique Raspberry pi v3
- Étape 4 - Configuration de la carte électronique Arduino Uno
- Étape 5 - Branchement des composants
- Étape 6 - Branchement des servo-moteurs
- Étape 7 - Test des codes
- Étape 8 - Assemblage
- Étape 9 - Test final
- Commentaires



Matériaux

Matériaux pour le boîtier

- Planche en bois 3mm d'épaisseur / 30cm de large / 60 cm de long
- Plastique pour découpeuse laser (ici du PLA) environ (100g)

Composants électroniques

- 1x Raspberry pi 3
- 6x Servo moteurs linéaires
- 1x Mini Haut parleur (Module haut-parleur SKU00101)
- 1x Cordon Jack CA35M
- 1x Carte Arduino Uno
- 5x boutons poussoirs
- 1x Capteur Ultrason
- 1x Caméra compatible Raspberry pi disponible chez Kubii
- 40x Fils de connexion électronique

Outils

- Découpeuse laser
- Imprimante 3D (ici la Ultimaker 2+)
- Ordinateur

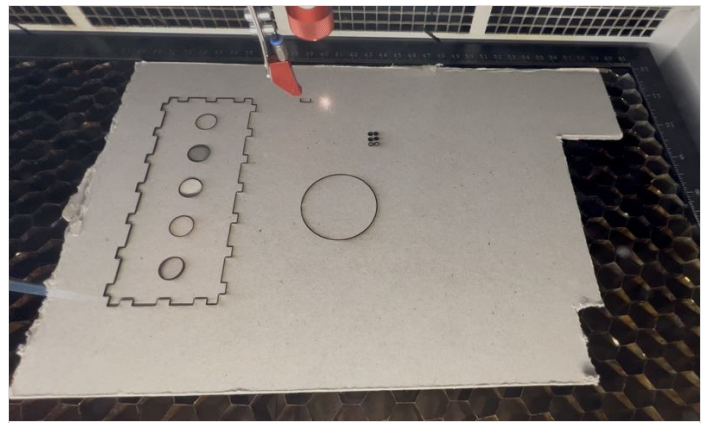




- support servomoteur .stl
 - joint servomoteur .stl
 - lettre braille 1.stl
 - lettre braille 2.stl
 - lettre braille 3.stl
 - Piece_arriere_file.dxf
 - Piece_base_1.dxf
 - Piece_base_2.dxf
 - Piece_bouton_d.dxf
 - Piece_camera.dxf
 - Piece_enceinte.dxf
 - Arduino_code.ino
 - Raspberry_code.docx
-

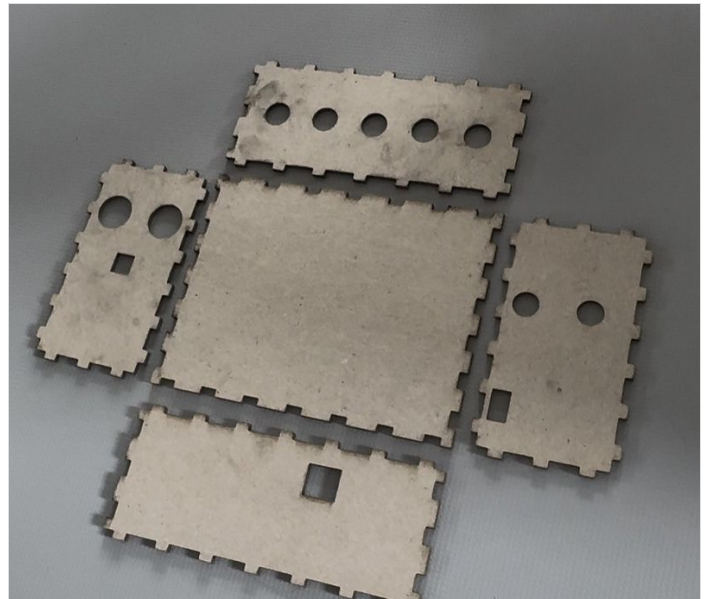
Étape 1 - Découpe des pièces en bois du boîtier

- Connecter l'ordinateur à la découpeuse laser.
- Ouvrir le logiciel Trotek (Nécessaire à la découpe).
- Ouvrir les pièces à découper en format dxf dans le logiciel.
- Optimiser l'espace sur la planche afin d'avoir les moins de perte de matière et déplaçant les pièces.
- Lancer la découpe.
- Nettoyer les pièces afin d'éviter des tâches dues au bois brûlé.



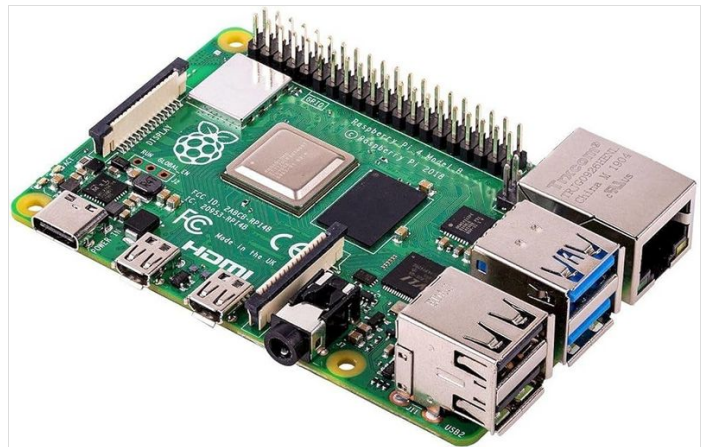
Étape 2 - Assemblage des pièces en bois

- Assembler les pièces découpées grâce aux images ci-contre.
- Les coller via de la colle à bois.



Étape 3 - Configuration de la carte électronique Raspberry pi v3

- Configurer votre carte Raspberry pi.
- Alimenter la carte Raspberry pi via son câble d'alimentation à l'ordinateur et en hdmi à l'écran.
- Copier le code "Raspberry_code" dans un fichier python au même nom.
- Avant de lancer ce fichier, vous devez d'abord créer et démarrer un environnement virtuel à l'aide du terminal. Cela permet de préparer un espace propre pour faire fonctionner votre programme.
- Dans le terminal, taper "python3 Raspberry_code.py".
- Le code devrait se lancer.
- Si c'est le cas, débrancher la carte raspberry pi, sinon regarder un tutoriel sur internet.



Étape 4 - Configuration de la carte électronique Arduino Uno

- Alimenter la carte Arduino Uno via son câble d'alimentation à ordinateur.
- Ouvrir Le logiciel Arduino IDE.
- Ouvrir le fichier "Arduino_code.ino".
- Uploader le code.
- Si le code parvient à s'uploader, débrancher la carte Arduino, sinon regarder un tutoriel sur internet.



```
Arduino_code.ino
1  #include <Wire.h>
2  #include <Servo.h>
3
4  Servo servos[6]; // Tableau de 6 servos
5
6  const int pinsServos[6] = {3, 5, 6, 9, 10, 11}; // Broches des servos
7
8  unsigned long tempsServo = 0;
9
10 bool servosActifs = false;
11
12 void setup() {
13
14   Serial.begin(9600); // Initialisation du moniteur série
15
16   Serial.println("✅ Arduino prêt, en attente des commandes I2C...");
17
18   Wire.begin(8); // Arduino en esclave I2C (adresse 8)
19
20   Wire.onReceive(recevoirCommande);
21
22   // Initialiser les servos à la position souhaitée (par exemple, 45°)
23
24   for (int i = 0; i < 6; i++) {
25
26     servos[i].attach(pinsServos[i]);
27
28     servos[i].write(45); // Position initiale (en bas)
29
30   }
31
32
33
34   void loop() {
35
36     if (servosActifs && millis() - tempsServo >= 3000) {
37
38       Serial.println("🔄 Retour des servos à la position initiale.");
39
40       for (int i = 0; i < 6; i++) {
41
42         servos[i].write(45); // Retour à la position initiale (en bas)
```

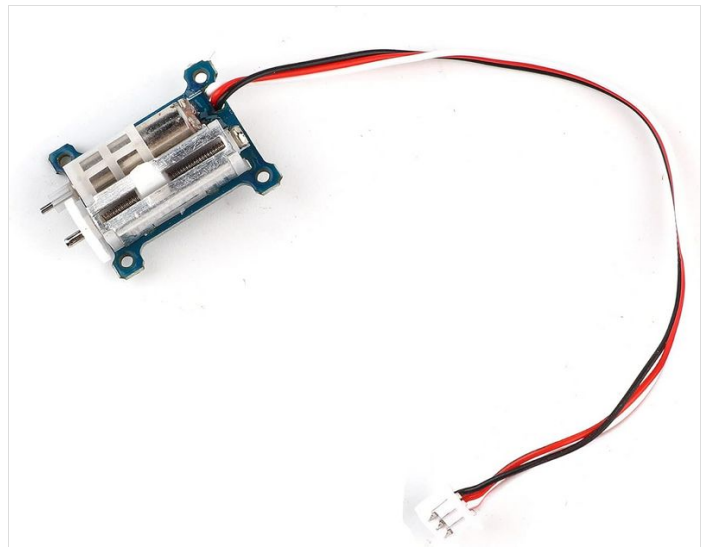
Étape 5 - Branchement des composants

- Une fois les 2 cartes débranchées, les connecter en I2C via 3 fils de connexion M>F.(SDA; SCL; GND)
- Brancher la caméra à la carte raspberry pi.
- Connecter le capteur de distance via 4 fils de connexion F>F à la carte Raspberry pi(VCC; GND; TRIG; ECHO)
- Connecter les boutons sur les GPIO {17; 22; 23; 24; 27} de la Raspberry pi et les alimenter via la carte Arduino
- Connecter le haut-parleur via 2 fils de connexion F>F.(VCC; GND) et le câble jack



Étape 6 - Branchement des servo-moteurs

-Connecter en PWM les 6 servo-moteurs linéaires à la carte Arduino en utilisant son alimentation et via les ports {3; 5; 6; 9; 10; 11}.



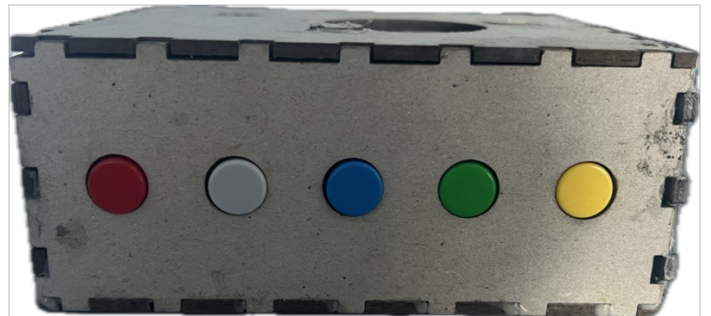
Étape 7 - Test des codes

-Alimenter les 2 carte électroniques.

-Attendre quelques instants que la raspberry se lance.

-Tester les boutons:

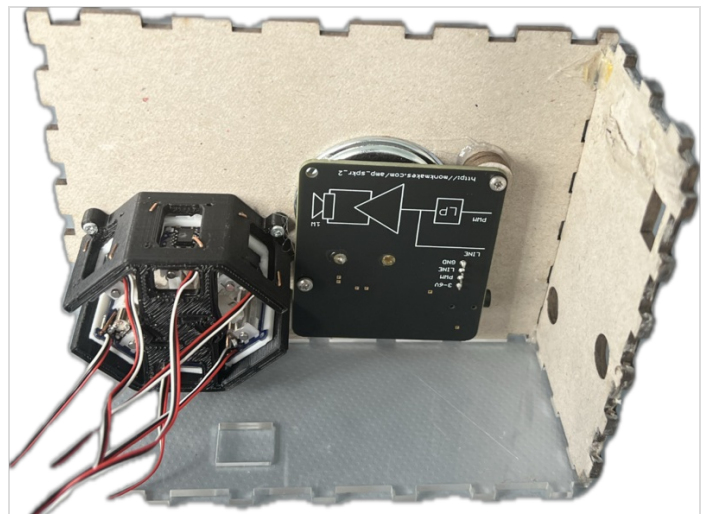
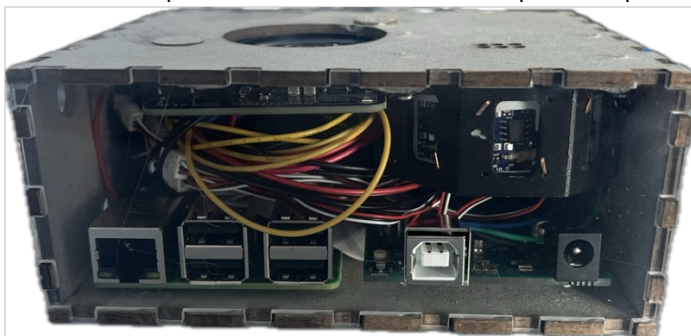
- Bouton 1: Début du mode leçon (Lettre dans l'ordre alphabétique)
- Bouton 2: Début du mode exercice (Lettre dans le désordre)
- Bouton 3: Lancement de la caméra:
 - Positionner le capteur de distance à environ 15cm, +/- 1cm d'un texte
 - Ne plus bouger
 - Attendre quelques secondes que la raspberry pi reconnaisse des lettres
- Bouton 4: Lettre suivante(Du texte détecté via la caméra)
- Bouton 5: Lettre précédente (Du texte détecté via la caméra)



Étape 8 - Assemblage

-Assembler les composants dans le boîtier afin de les protéger.

-Utiliser des vis pour un meilleur maintien des composants si possible.



Étape 9 - Test final

-Tester les boutons et valider les actions engendrées.

