

Piano automate

Ce projet consiste en l'automatisation d'un mini-piano grâce à 15 servomoteurs alignés sur les touches, contrôlés par un Arduino. Chaque servomoteur actionne une touche, permettant de jouer des mélodies programmées de façon électronique. C'est une fusion entre électronique, programmation et musique DIY.



Difficulté **Moyen**



Durée **5 heure(s)**



Catégories **Art, Électronique, Musique & Sons, Jeux & Loisirs, Robotique**



Coût **20 EUR (€)**

Sommaire

Introduction

Étape 1 - le câblage et alimentation

Étape 2 - l'impression 3D

Étape 3 - téléverser le code

Étape 4 - Téléverser le code arduino

Commentaires

Introduction

Ce projet automatise un mini-piano avec 15 servomoteurs contrôlés par un Arduino. Chaque servomoteur actionne une touche, permettant de jouer des mélodies programmées. Le tutoriel couvre le matériel, le câblage, l'alimentation, l'impression 3D des supports et le téléversement du code.



Matériaux

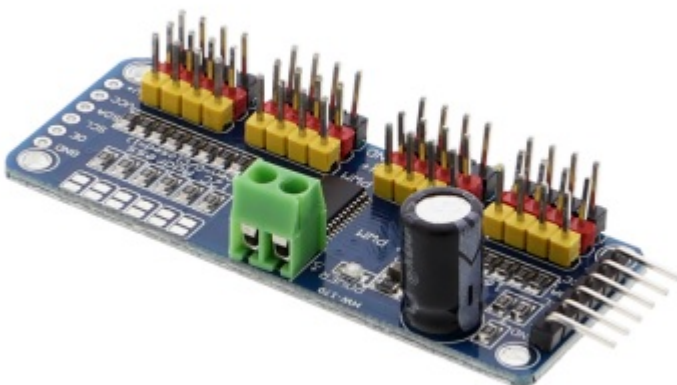
Matériel nécessaire

- Arduino Uno
- PCA9685 (module pilote de servomoteurs)
- 15 servomoteurs
- Câble USB-A
- Câbles Dupont
- Alimentation 6V
- Supports imprimés en 3D pour servomoteurs
- Vis et écrous pour fixation
- Colle ou ruban adhésif double-face (optionnel, pour maintenir certains éléments)

Outils

Outils nécessaires

- Ordinateur (pour programmer l'Arduino)
- Logiciel Arduino IDE
- Imprimante 3D (pour les supports des servomoteurs)
- Tournevis (pour vis et écrous)
- Pince coupante / pince à dénuder (pour câbles Dupont)
- Multimètre (optionnel, pour vérifier alimentation et câblage)
- Fer à souder et étain (optionnel, si certains câbles nécessitent une soudure)





Étape 1 - le câblage et alimentation

- **Alimentation des servomoteurs**

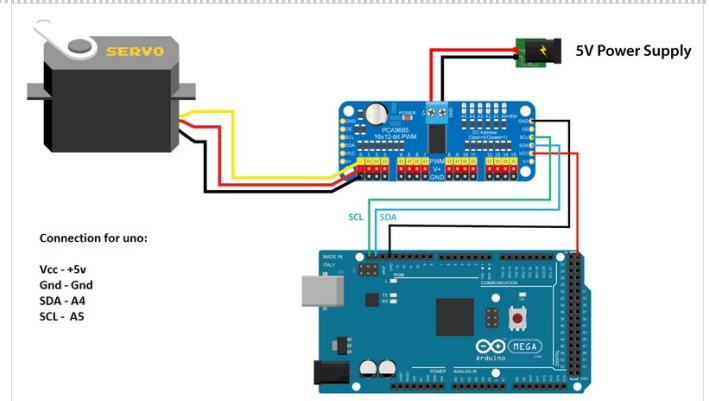
- Connectez le **+6V** de votre alimentation externe à la broche **V+** du PCA9685.
- Connectez la **masse (GND)** de l'alimentation à la broche **GND** du PCA9685.

- **Connexion I2C PCA9685 → Arduino**

- Reliez la broche **SDA** du PCA9685 à la broche **A4** de l'Arduino Uno.
- Reliez la broche **SCL** du PCA9685 à la broche **A5** de l'Arduino Uno.
- Reliez également la **GND** du PCA9685 à la **GND** de l'Arduino pour partager la masse.

- **Connexion des servomoteurs au PCA9685**

- Chaque servomoteur a trois fils : **VCC (+)**, **GND (-)** et **signal**.
- Branchez **VCC** à **V+** du PCA9685, **GND** à **GND** du PCA9685.
- Branchez le fil **signal** de chaque servomoteur à un canal de sortie du PCA9685



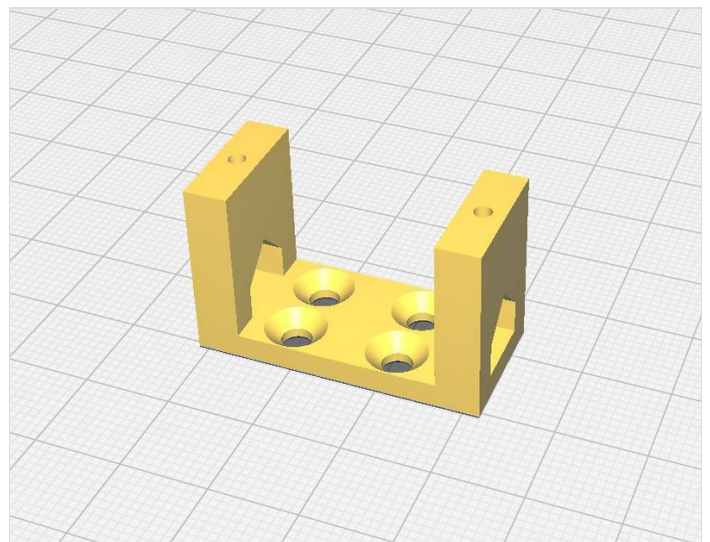
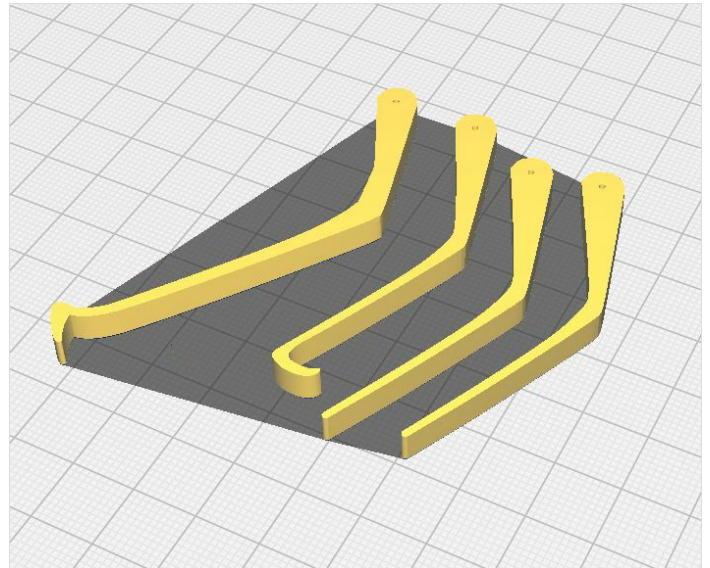
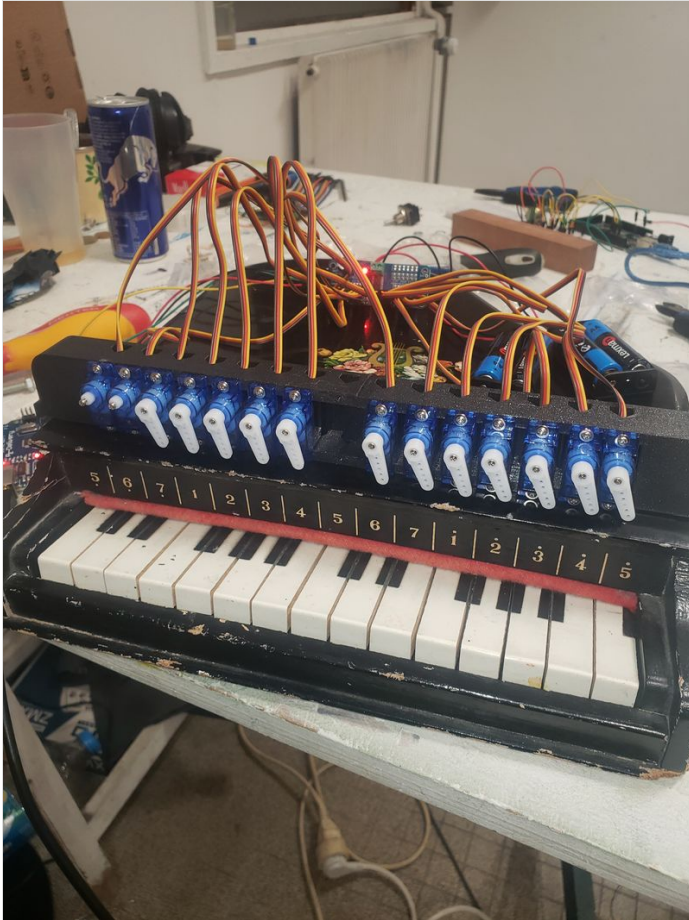
⚠ ...L'alimentation des servomoteurs est critique : ne les alimentez jamais directement depuis l'Arduino, car cela pourrait endommager la carte. Utilisez toujours une source externe 6V capable de fournir suffisamment de courant pour tous les servomoteurs. Vérifiez que toutes les masses (GND) sont reliées entre elles pour éviter des différences de potentiel pouvant provoquer des comportements imprévisibles ou des dommages électroniques. Débranchez l'alimentation avant toute modification du câblage et évitez les courts-circuits lors de l'assemblage.

Étape 2 - l'impression 3D

Ouvrez le fichier **STL** des supports dans un slicer comme **Cura**, configurez vos paramètres d'impression, générez le **G-code** et imprimez. Vérifiez que les pièces s'adaptent correctement aux servomoteurs avant le montage.

La forme de votre axe de servomoteurs dépendra du piano donc il vaut mieux prendre une pièce .Stl et la multiplier suivant le nombres de moteurs.

Voir lien pour model 3D en open source ici .



Étape 3 - téléverser le code

```
1 #include <Wire.h> // Bibliothèque pour la communication I2C
2 #include <Adafruit_PWMServoDriver.h> // Bibliothèque pour piloter le PCA9685 (servomoteurs)
3
4 // Création d'une instance du module PCA9685 à l'adresse 0x40
5 Adafruit_PWMServoDriver pca = Adafruit_PWMServoDriver(0x40);
6
7 // Valeurs PWM correspondant aux positions repos et levée des servos
8 int SERVOMIN = 100; // Position repos
9 int SERVOMAX = 200; // Position levée complète
10
11 // Valeur pour un mouvement plus puissant (~40°)
12 int SERVO_FRAPPE = SERVOMIN + ((SERVOMAX - SERVOMIN) / 4); // Position intermédiaire pour frapper
13
14 // Délais utilisés pour frapper et revenir
15 #define DELAI_FRAPPE 200 // Durée pour lever le bras du servo
16 #define DELAI_RETOUR 250 // Durée pour revenir à la position repos
17 #define DELAI_SECURITE 50 // Petite pause pour éviter chevauchement des frappes
18
19 // Durées des notes pour le rythme de la musique
```

```

20 #define NOIRE 350      // Durée d'une noire
21 #define CROCHE 175    // Durée d'une croche
22 #define BLANCHE 700   // Durée d'une blanche
23
24 // Mapping des notes sur les 15 servos (de DO grave à DO aigu)
25 #define DO 0
26 #define RE 1
27 #define MI 2
28 #define FA 3
29 #define SOL 4
30 #define LA 5
31 #define SI 6
32 #define DO2 7
33 #define RE2 8
34 #define MI2 9
35 #define FA2 10
36 #define SOL2 11
37 #define LA2 12
38 #define SI2 13
39 #define DO3 14
40
41 // ----- SETUP -----
42 void setup() {
43   Serial.begin(115200);      // Initialisation du port série pour debug
44   Serial.println("Piano 15 touches - Ne Pleure Pas Jeannette");
45
46   pca.begin();              // Démarrage du module PCA9685
47   pca.setPWMFreq(50);       // Fréquence standard servomoteur (50Hz)
48   delay(100);
49
50   // Initialisation des servos à la position repos
51   Serial.println("Initialisation des servos...");
52   for (int i = 0; i < 15; i++) {
53     pca.setPWM(i, 0, SERVOMIN);      // Met chaque servo à la position repos
54     delay(50);                      // Petite pause pour éviter surcharge I2C
55   }
56
57   delay(1000);
58   Serial.println("Pret! Demarrage dans 2 secondes...");
59   delay(2000);                      // Pause avant le début
60 }
61
62 // ----- LOOP PRINCIPAL -----
63 void loop() {
64   jouerNePleurePasJeannette();      // Appel de la fonction qui joue la musique complète
65   delay(3000);                      // Pause entre les répétitions de la chanson
66 }
67
68 // ----- FONCTIONS DE BASE -----
69
70 // Frapper une touche spécifique
71 void frapperTouche(int servoNum) {
72   if (servoNum < 0 || servoNum >= 15) return; // Vérifie que le servo existe
73
74   Serial.print("Note: Servo ");
75   Serial.println(servoNum);         // Affiche dans le moniteur série pour debug
76
77   pca.setPWM(servoNum, 0, SERVO_FRAPPE); // Lever le bras du servo pour frapper
78   delay(DELAI_FRAPPE);
79
80   pca.setPWM(servoNum, 0, SERVOMIN);   // Revenir à la position repos
81   delay(DELAI_RETOUR);
82   delay(DELAI_SECURITE);              // Pause pour sécurité
83 }
84
85 // Jouer une note avec sa durée
86 void jouerNote(int note, int duree) {
87   frapperTouche(note);               // Frapper la touche
88   int tempsRestant = duree - (DELAI_FRAPPE + DELAI_RETOUR + DELAI_SECURITE);
89   if (tempsRestant > 0) {
90     delay(tempsRestant);             // Attendre le reste de la durée de la note
91   }
92 }
93

```



```

94 // Silence / pause
95 void silence(int duree) {
96     delay(duree);                // Simple pause pour un silence
97 }
98
99 // ----- MUSIQUE : "Ne pleure pas Jeannette" -----
100 void jouerNePleurePasJeannette() {
101     Serial.println("=== Ne pleure pas Jeannette - Version Etendue ===");
102
103     // Introduction - graves (servos 0 à 4)
104     jouerNote(DO, CROCHE);
105     jouerNote(MI, CROCHE);
106     jouerNote(SOL, CROCHE);
107     jouerNote(DO2, CROCHE);
108
109     // Partie principale - octave moyenne
110     jouerNote(SOL, CROCHE);
111     jouerNote(SOL, CROCHE);
112     jouerNote(LA, CROCHE);
113     jouerNote(SI, CROCHE);
114     jouerNote(DO2, NOIRE);
115     jouerNote(SI, NOIRE);
116
117     // "Tra la la" - monte vers aigu
118     jouerNote(LA, CROCHE);
119     jouerNote(SOL, CROCHE);
120     jouerNote(LA, CROCHE);
121     jouerNote(SI, CROCHE);
122     jouerNote(DO2, NOIRE);
123     jouerNote(RE2, NOIRE);
124
125     // Octave haute (servos 7-14)
126     jouerNote(DO2, CROCHE);
127     jouerNote(RE2, CROCHE);
128     jouerNote(MI2, CROCHE);
129     jouerNote(FA2, CROCHE);
130     jouerNote(SOL2, NOIRE);
131     jouerNote(FA2, NOIRE);
132
133     // Descente rapide
134     jouerNote(MI2, CROCHE);
135     jouerNote(RE2, CROCHE);
136     jouerNote(DO2, CROCHE);
137     jouerNote(SI, CROCHE);
138     jouerNote(LA, NOIRE);
139     jouerNote(SOL, NOIRE);
140
141     silence(300);                // Petite pause
142
143     // Interlude - arpège complet
144     jouerNote(DO, CROCHE);
145     jouerNote(RE, CROCHE);
146     jouerNote(MI, CROCHE);
147     jouerNote(FA, CROCHE);
148     jouerNote(SOL, CROCHE);
149     jouerNote(LA, CROCHE);
150     jouerNote(SI, CROCHE);
151     jouerNote(DO2, CROCHE);
152     jouerNote(RE2, CROCHE);
153     jouerNote(MI2, CROCHE);
154     jouerNote(FA2, CROCHE);
155     jouerNote(SOL2, CROCHE);
156     jouerNote(LA2, CROCHE);
157     jouerNote(SI2, CROCHE);
158     jouerNote(DO3, NOIRE);      // Note la plus haute
159
160     silence(300);                // Petite pause
161
162     // Refrain
163     jouerNote(SI, CROCHE);
164     jouerNote(SI, CROCHE);
165     jouerNote(DO2, CROCHE);
166     jouerNote(RE2, CROCHE);
167     jouerNote(MI2, NOIRE);

```

```

168 jouerNote(RE2, NOIRE);
169
170 // Descente
171 jouerNote(DO2, CROCHE);
172 jouerNote(SI, CROCHE);
173 jouerNote(LA, CROCHE);
174 jouerNote(SOL, CROCHE);
175 jouerNote(FA, NOIRE);
176 jouerNote(MI, NOIRE);
177
178 // Finale sur toute l'étendue
179 jouerNote(SOL2, CROCHE);
180 jouerNote(FA2, CROCHE);
181 jouerNote(MI2, CROCHE);
182 jouerNote(RE2, CROCHE);
183 jouerNote(DO2, CROCHE);
184 jouerNote(SI, CROCHE);
185 jouerNote(LA, CROCHE);
186 jouerNote(SOL, CROCHE);
187 jouerNote(FA, CROCHE);
188 jouerNote(MI, CROCHE);
189 jouerNote(RE, CROCHE);
190 jouerNote(DO, NOIRE); // Retour grave
191
192 silence(NOIRE); // Petite pause
193
194 // Final - accord rapide (3 notes simultanées)
195 jouerNote(DO, CROCHE);
196 delay(30);
197 jouerNote(MI, CROCHE);
198 delay(30);
199 jouerNote(SOL, CROCHE);
200 delay(30);
201 jouerNote(DO2, NOIRE);
202
203 Serial.println("=== Fin ===");
204 }

```

Étape 4 - Téléverser le code arduino

- **Installer le logiciel Arduino IDE** Téléchargez et installez l'**Arduino IDE** sur votre ordinateur (Windows / macOS / Linux). Cela vous permettra d'écrire, de vérifier et de téléverser votre code.
- **Ouvrir votre code** Lancez l'IDE Arduino et ouvrez votre fichier **.ino** contenant le code de votre piano automatisé.
- **Connecter l'Arduino** Branchez votre **Arduino Uno ou clone** à l'ordinateur avec un **câble USB-A** (assurez-vous que c'est un **câble de données**, pas seulement de charge).
- **Sélectionner la carte et le port**
 - Dans le menu **Outils → Carte**, choisissez **Arduino Uno** (ou la version compatible de votre clone).
 - Ensuite, allez dans **Outils → Port** et sélectionnez le port COM ou USB auquel votre carte est connectée.
- **Vérifier le code** Avant de téléverser, cliquez sur le bouton **Vérifier** (icône avec une coche) pour compiler le code et détecter d'éventuelles erreurs.
- **Téléverser le code** Appuyez sur le bouton **Téléverser** (icône avec une flèche vers la droite). L'IDE va compiler puis envoyer automatiquement le code à la carte. Les LED **RX / TX** de l'Arduino clignoteront pendant l'envoi, puis votre sketch commencera à s'exécuter une fois le téléversement terminé.



