

SCHAEFFER

Quentin

**Synthèse personnelle du**

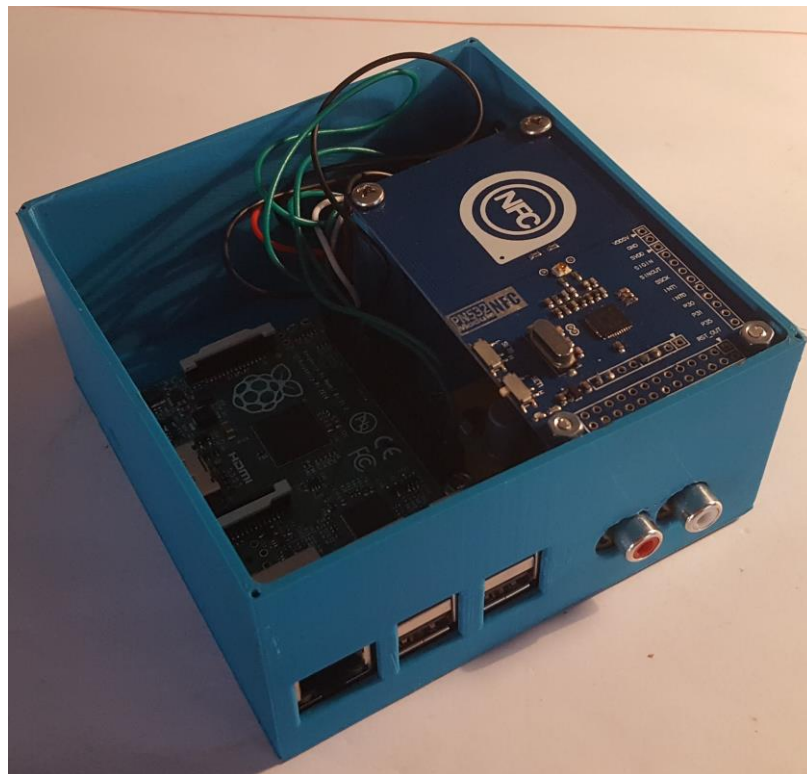
**Projet Interdisciplinaire**



Noms des autres élèves du groupe : MATHIEU Arthur, DUBOURDIEU Aurélien, SCHEFFER Jérémy

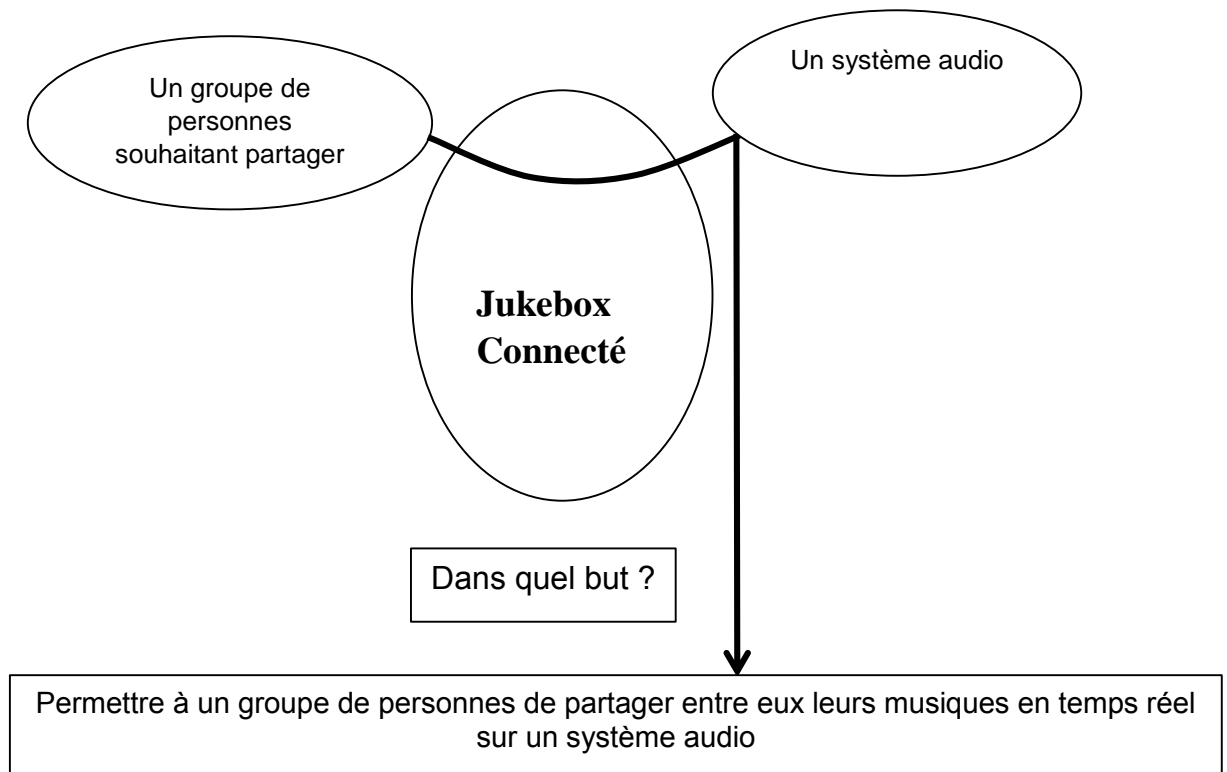
Sujet d'étude : Créer un « jukebox » connecté

Intitulé du sujet : Créer un « jukebox » connecté afin de permettre à un groupe de partager entre eux leurs musiques en temps réel sur un système audio.

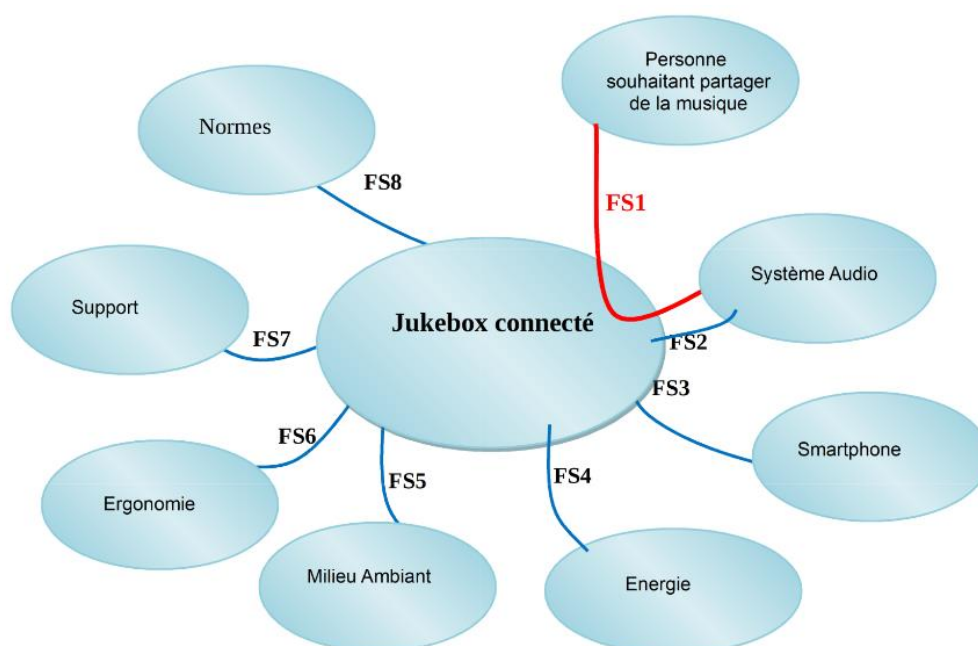


### Notre démarche commune :

Pendant la première partie de cette étude, nous avons réfléchi tous ensemble sur les solutions pour créer ce jukebox. Tout d'abord, nous avons validé le besoin en nous demandant si ce besoin est vraiment réel et les risques qu'il a de disparaître prochainement.



Après avoir clairement énoncé le besoin, nous avons étudié toutes les interactions et fonctions de notre système avec l'extérieur schématisés sur le diagramme pieuvre ci-dessous :



Nous avons identifié les fonctions de service suivantes :

**FS1** : Permettre à toute personne de partager sa musique sur un système audio

**FS2** : Être connecté à un système audio

**FS3** : Être contrôlé à distance depuis un smartphone

**FS4** : Être alimenté en énergie électrique

**FS5** : Résister aux agressions du milieu ambiant

**FS6** : Être simple d'utilisation

**FS7** : Être stable sur son support

**FS8** : Respecter les normes en vigueur

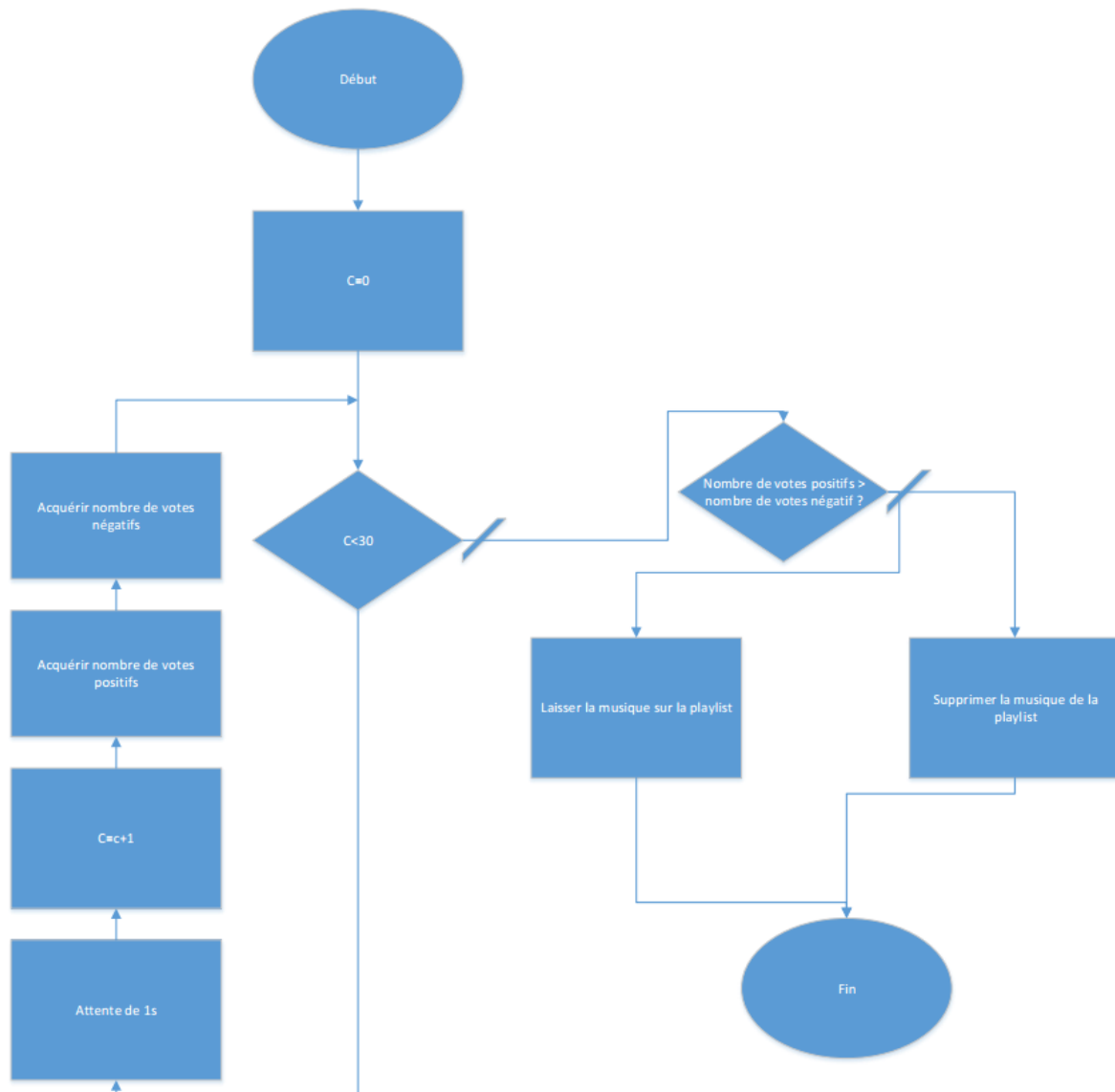
Ensuite nous avons définis chaque fonction de manière détaillée pour nous donner un cadre de travail :

Fonctions	Désignation	Critères d'appréciation	Niveau	Flexibilité		Observations
				Limite d'acceptation	Taux	
<b>FS1</b>	Permettre à toute personne de partager sa musique sur un système audio.	- Mise en marche	Programme adapté	/		Dès la mise sous tension.
		- Sélection	Algorithme	/		Gestion de la priorité de lecture avec un algorithme favorisant la répartition des musiques jouées par les utilisateurs.
		- Vote - Choix - Temps de requête		Maximum		
<b>FS2</b>	Être connecté au système audio.	- Type de connexion :	RCA	/		Les enceintes pourront être connectées, au choix, en RCA, jack 3.5 ou en Bluetooth.
			Bluetooth	/		
<b>FS3</b>	Être contrôlé à distance depuis un smartphone.	- Dialogue avec l'interface	Via la page Web	/		Connexion aisée à l'aide du NFC et du QR Code.
		- Mode de connexion à l'interface	QR Code ; NFC ; IP	/		
		- Connectivité	Wifi (portée: 8 mètre) ; Ethernet	Minimum		
<b>FS4</b>	Être alimenté en énergie électrique.	- Alimentation	230V 50 Hz	/		/
<b>FS5</b>	Résister aux agressions du milieu ambiant.	- Aux chocs	Chute de 80cm	± 20cm		/

<b>FS6</b>	Être simple d'utilisation.	- Interface - Réactivité	Simple 1s	/ Maximum		L'utilisation doit pouvoir se faire sans explication de fonctionnement et avec une rapidité convenable.
<b>FS7</b>	Être stable sur son support.	- Stabilité	Support horizontal	$\pm 10^\circ$		Il devra être stable sur une surface horizontale.
<b>FS8</b>	Respecter les normes en vigueur.	- Normes françaises et européennes	/	/		/

### Travail personnel :

Pour ma part, je me suis tout d'abord occupé de créer et d'intégrer un système de vote dans Mopidy. Pour ce faire, j'ai tout d'abord créé un algorithme qui m'a permis de voir plus clairement comment je dois construire ce système de vote :



J'ai ensuite cherché différents moyen pour créer ce système de vote, notamment en incrémentant une variable pour le vote négatif dans le fichier JavaScript, pour ensuite l'envoyer au fichier Python, qui va passer la musique si le nombre de votes négatif est égal à celui requis dans les fichiers de configuration :

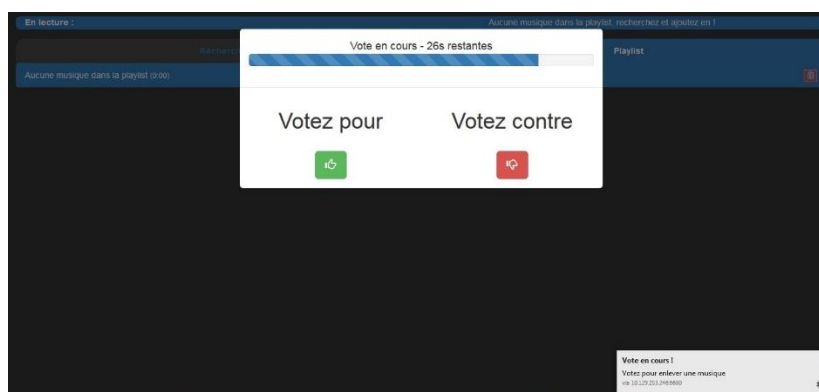
```
# Si la musique jouée est différente de celle enregistrée, effacer les votes
if (currentTrackURI != self.dataPositif["track"]):
    self.dataPositif["track"] = currentTrackURI
    self.dataPositif["votes_positifs"] = []

if (self.request.remote_ip in self.dataPositif["votes_positifs"]): # L'utilisateur a déjà voté
    self.write("Vous avez déjà voté pour passer cette chanson !")
else: # le vote est valide
    self.dataPositif["votes_positifs"].append(self.request.remote_ip)
    if (len(self.dataPositif["votes_positifs"]) == self.requiredVotes):
        self.core.playback.next()
        self.write("Changement de musique...")
    else:
        self.write("Vous avez voté pour garder cette musique.")
```

Mais, cette technique avait l'inconvénient de n'avoir qu'un vote négatif et pas de vote positif, mais également elle ne prend pas en compte le nombre de personnes qui ont voté, elle fonctionne sur un nombre fixe de vote pour passer la musique.

Je me suis donc tourné vers une autre façon de gérer ce vote. L'utilisateur appuiera sur un bouton sur l'interface web, ce qui déclenche un minuteur de 30sec, période durant laquelle tous les utilisateurs pourront voter pour passer la musique. L'appuie sur un bouton de vote (positif ou négatif) déclenche un script PHP qui va lire la valeur de la variable de vote (positif ou négatif) dans un fichier texte, qui va l'incrémenter et la réécrire dans le fichier texte. Ensuite, au bout des 30 secondes de vote, le JavaScript ira rechercher les valeurs des votes positifs et négatifs, et si les votes négatifs sont plus nombreux que les votes positifs, la musique ne sera pas jouée.

L'interface de vote :

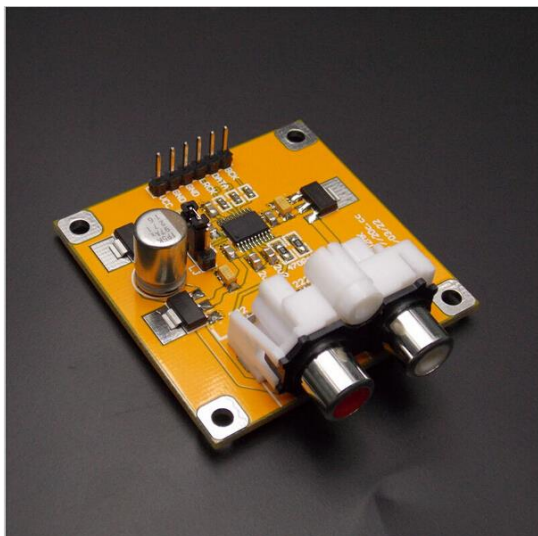


Le script PHP :

```
1  <?php
2  //on ouvre le fichier
3  $fichier_resultats=fopen('result_negatif.txt', 'r+');
4  //on recupere la variable
5  $vote_n=fgets($fichier_resultats);
6  //on efface le contenu du fichier
7  ftruncate($fichier_resultats,0);
8  //on se place au debut du fichier
9  fseek($fichier_resultats,0);
10 //on incremente la variable
11 $vote_n=$vote_n+1;
12 //on ecrit les variables dans le fichier txt
13 fputs($fichier_resultats, $vote_n);
14 //on ferme le fichier
15 fclose($fichier_resultats);
16 //on ferme l'onglet
17 echo "<script language='javascript'>window.close()</script>";
18 ?>
19 <html>
20 <h1>Le vote negatif a ete enregistre</h1>
21 </html>
```

Ensuite, la personne qui devait s'occuper des branchements des différents composants étant en manque de temps, je m'en suis également occupé. Nous avons décidé de brancher deux composants supplémentaire sur la Raspberry, notamment une carte son, celle intégrée dans la Raspberry étant de mauvaise qualité. Nous avons également décidé d'ajouter une carte NFC, de manière à rendre la connexion à la Raspberry plus facile. Nous avons également pensé à ajouter un bouton d'arrêt pour la Raspberry, avec une LED confirmant qu'elle est bien allumée / éteinte, afin d'éteindre correctement la Raspberry et éviter tous dysfonctionnement provoqué par un débranchement direct. Pour que les composants fonctionnent correctement, il faut bien évidemment les brancher sur les bons pins de la Raspberry mais également une configuration logicielle des cartes, au moyen de lignes de commande. Voici les différents composants choisis :

-La carte son :



-La carte NFC :



-Le bouton pour éteindre la Raspberry :



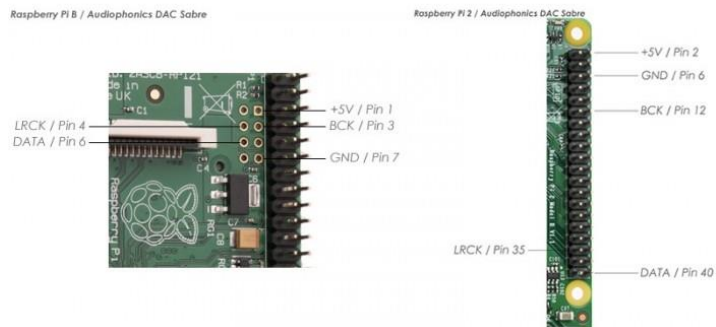
-La LED qui permet de savoir si la Raspberry est allumée :



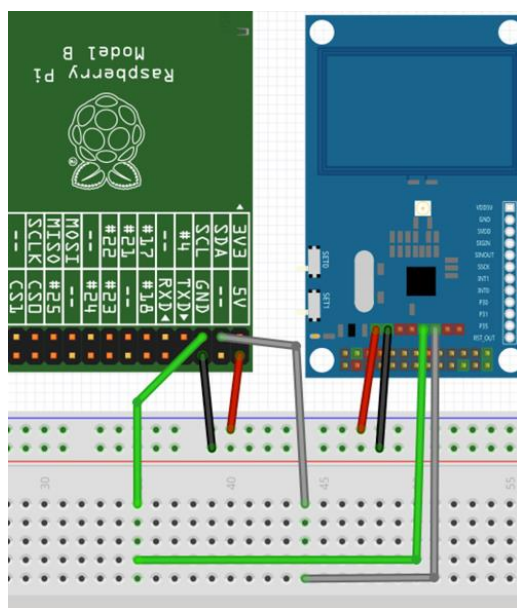


Voici maintenant les différents schémas de branchement des composants, nous avons décidé de connecter les différents composants ensemble au moyen de câbles soudés sur les différentes cartes.

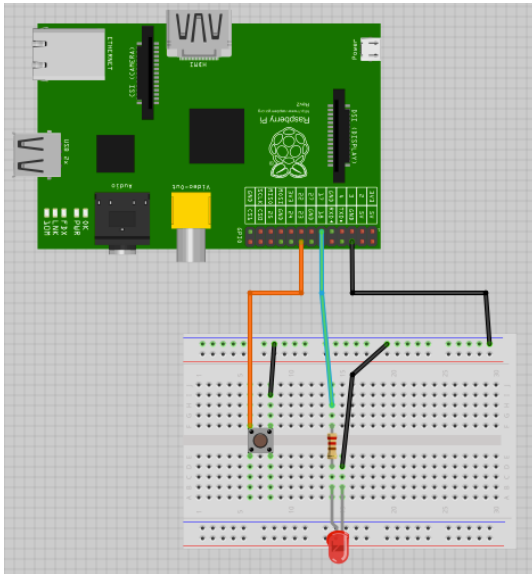
-La carte son :



-La carte NFC :

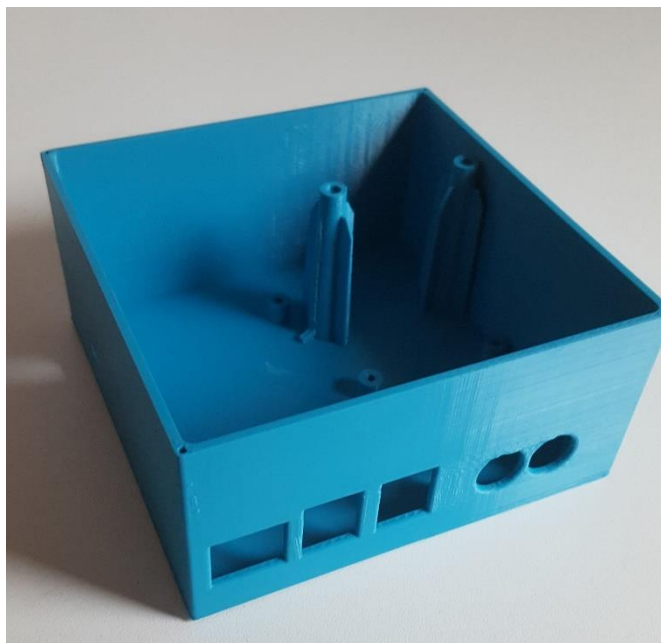


-Le bouton d'arrêt ainsi que la LED :



La personne qui devait s'occuper de créer le boîtier ayant eu quelques soucis pour réaliser ce dernier, avec Arthur, nous avons réalisé en dernière minute un nouveau boîtier que nous avons ensuite fait imprimer en 3D.

Voici le boîtier :



Je peux conclure que ce projet m'a appris beaucoup de choses, notamment le travail en groupe, mais également la programmation dans différents langages, notamment le PHP, l'HTML, le JavaScript. J'ai également appris à utiliser le Raspberry.

Nous remercions tous nos professeurs qui nous ont encadrés durant ce projet, tout particulièrement M.Selzer et M.Boinet, mais également le Lycée du Haut Barr pour nous avoir prêté toutes les infrastructures qui nous ont permis de mener à bien ce projet.