

Antonin Segault
Cédric Lanoir

Projet de fin d'études



Cahier de spécification

Master 2 PSM – 2012/2013
UFR STGI – Université de Franche Comté

Sommaire

Introduction.....	3
Hardware.....	4
Capteurs.....	4
Température.....	4
Lumière.....	4
Distance.....	4
Commande.....	5
Schéma électronique.....	5
Design hardware.....	7
Software.....	8
Interaction série avec Arduino	8
Interaction PureData / VST.....	9
Système quadriphonique.....	11
Design software.....	11
Système de présets	14
Installation.....	15
Disposition optimale du Sy~.....	15
Schéma d'installation.....	16
Principe d'étalonnage de la luminosité.....	16
Mise en route.....	17
Posters et expositions.....	17
Gestion de projet.....	18
Tableau des tâches.....	18
Diagramme de Gantt.....	20

Introduction

Etudiants du parcours Musique et Son du Master Produits et Services Multimédia, nous sommes tous deux musiciens et passionnés par les formes artistiques avant-gardistes. Lorsque Jean-François Charles, professeur au Conservatoire de Montbéliard et intervenant au Département Multimédia et Informatique, nous présenta le sujet "Lutherie musicale - Le Syhamo", notre intérêt fut immédiat.

A travers ce Projet de Fin d'Etudes, nous souhaitons enrichir nos compétences techniques, en développement multimédia, électronique et musique assistée par ordinateur, et élargir notre horizon artistique. Nous espérons que nos compétences variées et nos profils complémentaires (orientation professionnelle pour Cédric et recherche pour Antonin) contribuerons à la réussite de ce projet.

"Une fois que l'on a goûté au Syhamo, il est impossible de vivre sans".

- Oscar Wiggli - 2012

Dans ce document, nous exposerons l'avancée de nos recherches artistiques et techniques. Nous présenterons et justifierons les choix que nous avons fait en fonction du cahier des charges. Enfin, nous décrirons l'organisation de la phase de production durant les trois prochains mois.

Hardware

Capteurs

Température

Le SYHAMO original comporte quatre thermistances, disposées de manière à ce que les spectateurs puissent les manipuler.

Les thermistances sont très peu chères, mais également peu précises (pour la mesure d'une température exacte) et plutôt lentes. Cependant, nous avons pu observer que leur vitesse de variation (de l'ordre de quelques secondes) n'est pas un véritable problème. Au contraire, elle permet plus de nuances pour celui qui les manipule. Par ailleurs, la température exacte n'a pas vraiment d'importance, seule compte la variation, et la chaleur des doigts suffit à obtenir des effets importants.

Lumière

Sur le SYHAMO, ce sont des photo-résistances qui mesurent les variations de la luminosité. Disposées sur la face avant de l'appareil, elles sont également à disposition du public.

Nous avons, à un moment, pensé utiliser des photodiodes, plus rapides mais plus chères. Cependant, l'expérience nous a montré que la vitesse de réaction des photo-résistances n'est pas réellement gênante.

Distance

Pour mesurer la distance des spectateurs jusqu'à 7 mètres, Oscar et Janine Wigglé ont choisi des capteurs à ultra-son (la portée des infrarouges ne dépassant pas un mètre). Trois de ces capteurs sont montés en façade du SYHAMO, pour mesurer la présence d'individus dans différentes zones.

Nous avons comparé les (nombreux) modèles existants et leurs caractéristiques. Nous cherchions une portée proche de 7 mètres, et un rayon assez large, pour un prix raisonnable.

Modèle	Prix	Portée	Résolution	Taux de lecture	Sortie numérique ?
LV-EZ0	24,25 €	6,45 m	25 mm	20 Hz	oui
HRLV-EZ0	32,16 €	5,00 m	1 mm	10 Hz	oui
XL-EZ0	46,95 €	7,65 m	10 mm	10 Hz	oui
XL-AE0	46,95 €	7,65 m	10 mm	10 Hz	non

Comparatif des modèles avec leurs spécifications

Après avoir effectué ce comparatif, nous avons remarqué qu'il existait deux modèles qui correspondaient à nos attentes en terme de prix/portée : XL-EZ0 et XL-AE0, la résolution de 1cm étant largement suffisante pour notre application. Nous avons finalement choisi le modèle **XL-EZ0** du fait de sa sortie numérique (indispensable comme nous l'expliquerons plus loin).

Jean-François Charles nous a alors fait remarquer qu'utiliser trois capteurs pointés dans la même direction n'apporte pas plus d'information qu'un seul capteur. Nous avons donc décidé de n'en utiliser que deux, pointés de manière à doubler la largeur de la zone couverte.

Commande

Le Conservatoire a pris en charge la commande des capteurs et des quelques autres composants nécessaires à la réalisation du circuit.

Schéma électronique

Les thermistances et photo-résistances voient leur résistance changer selon la valeur de la température et de la lumière. On peut donc récupérer leur mesure sous forme de variation de voltage (sur une entrée analogique) grâce à un pont diviseur de tension.

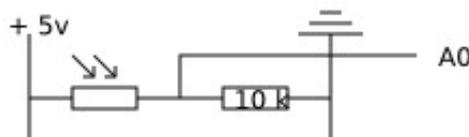


Schéma du pont diviseur de tension avec une photorésistance

Les capteurs sonars, eux, permettent de sortir les mesures sous forme numérique (Pulse Width Modulation sur la pin 2, par exemple). L'Arduino ne possédant que 6 entrées analogiques, occupées par les thermistances et photorésistances, c'est cette solution que nous avons retenue, car elle permet l'utilisation d'une entrée numérique.

Nous avons pensé inclure dans le circuit un système permettant de fixer le maximum de lumière, de manière à optimiser la plage de mesure. Un bouton poussoir a donc été ajouté de manière à permettre l'enregistrement de cette valeur.

Un autre bouton poussoir a été ajouté, de manière à passer d'une composition à l'autre, tandis qu'une LED (verte) indique que l'appareil est sous tension.

Voici donc le schéma électrique que nous avons conçu pour intégrer tous ces éléments :

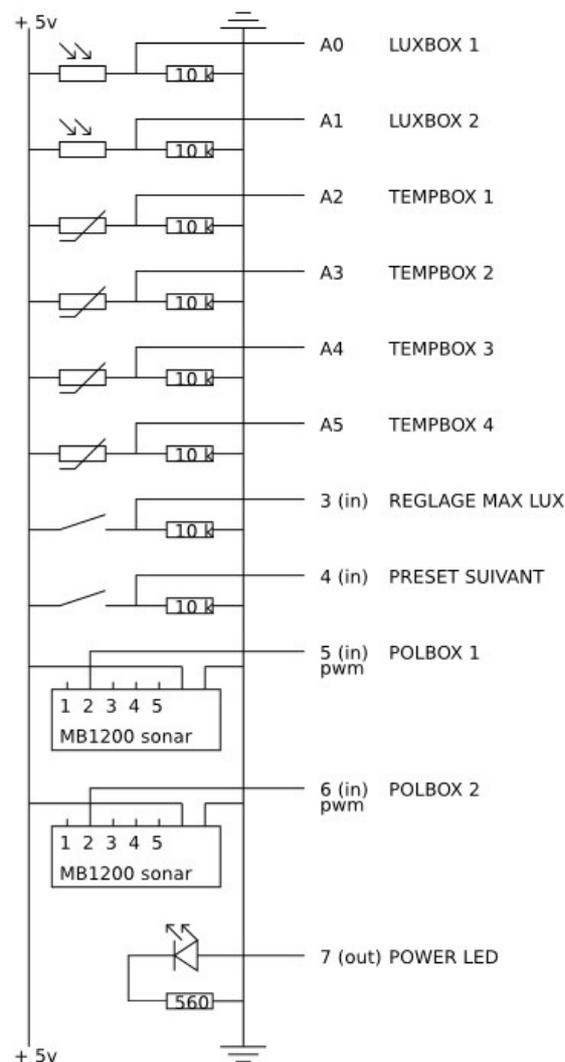


Schéma électrique complet

Design hardware

Prérequis matériel :

- 2 capteurs sonar orientés à l'horizontale
- 2 capteur de luminosité orientés à la verticale/diagonale
- 4 capteurs de température accessibles
- 1 bouton de preset accessible
- 1 LED de fonctionnement pas trop visible
- 1 bouton d'étalonnage pas trop visibles
- 1 câble USB pas trop visible

Prérequis fonctionnel :

- démontage possible sans impliquer un décablage, le circuit et les capteurs doivent donc être fixés sur un même support (pas les capteurs sur la boîte et le circuit sur le couvercle, par exemple)
- utilisabilité dans un maximum de lieux
- penser que le PC et la carte son ne devront pas être trop loin (1m environ) mais pourront être caché
- une hauteur d'environ 1m pour que les capteurs soient accessibles à tous (cette hauteur va notamment dépendre du support sur lequel le Sy~ sera posé)

Si les prérequis doivent être respectés nous sommes assez libre quant à la disposition générale et à l'esthétique du Sy~. Nous avons ainsi émis plusieurs propositions qui pourront être adaptées au fur et à mesure de l'avancée du projet.

Proposition 1 : Reprendre une forme globalement cubique (à l'instar du Syhamo) où tous les capteurs seraient disposés sur la face avant orientés à l'horizontale (capteurs sonar, capteurs de température) ou légèrement en hauteur (capteurs de luminosité). Les éléments qui ne demandent pas d'être visibles pourraient être situés à l'arrière de la « boîte » et il serait facile de disposer le bouton de preset sur le haut ou le côté de l'installation.

Cette proposition a l'avantage d'avoir une réalisation plutôt aisée et répond parfaitement aux prérequis matériel et fonctionnel mais elle demeure basique et peu originale. Une alternative à cette proposition serait de faire la face de devant convexe pour couper avec les angles droits et faciliter la captation sur un plus grand angle.

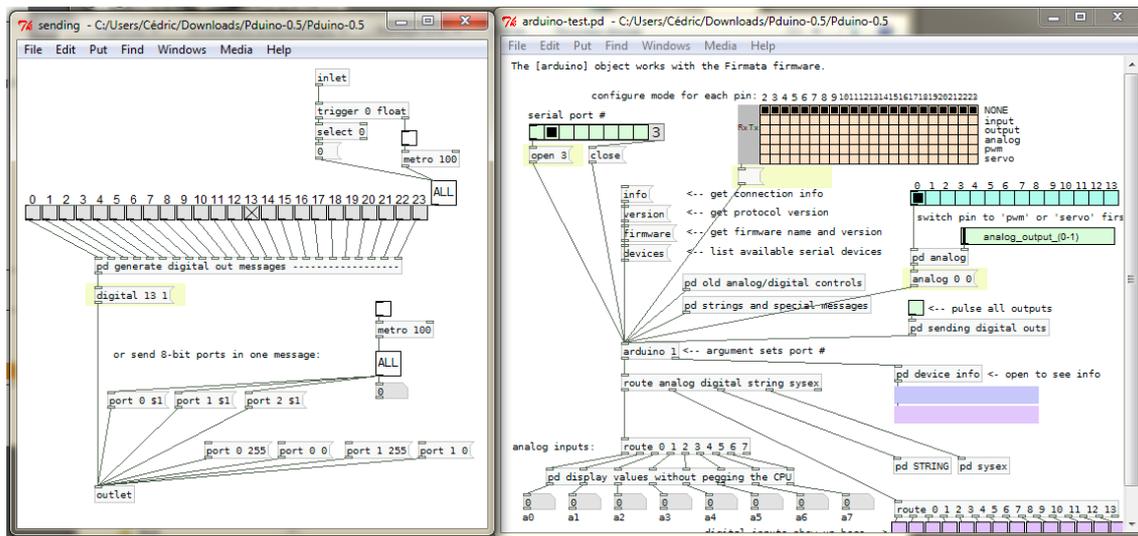
Proposition 2 : Partir sur l'idée d'une demi-sphère coupée à l'horizontale de manière à avoir une base plate et où tous les éléments seraient disposés sur la surface courbe. Cette proposition serait originale et intéressante en terme de conception mais également plus complexe notamment pour la matière première.

Software

Interaction série avec Arduino

Les récents cours de Arduino dans le parcours Musique et Son avec l'intervenant Jean-François Charles nous ont permis de mieux comprendre le fonctionnement des ports-séries et ainsi d'abandonner la librairie Firmata qui nous semblait être, alors, une des seules façons de permettre la communication entre Arduino et PureData.

Firmata proposait une interface complexe de gestion des ports-séries que nous estimons, à présent, être totalement superflue.



Interface de Firmata

Le développement Software de notre application s'en verra simplifié grâce à l'utilisation, basique, des ports-séries.

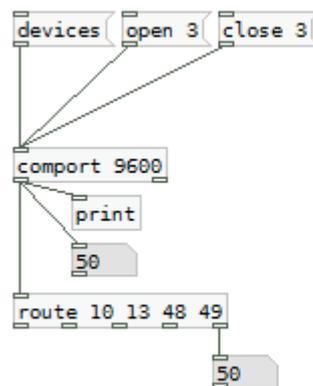
```

void setup(){
  // initialise la communication série
  Serial.begin(9600);
}
// boucle d'exécution
void loop(){
  // affichage dans le moniteur Arduino
  Serial.println(analogRead(A0));
  // écriture
  Serial.write(analogRead(A0)/2);

  // delai entre chaque boucle
  delay(10);
}

```

Code Arduino pour la communication en série



Patch PureData pour la communication en série

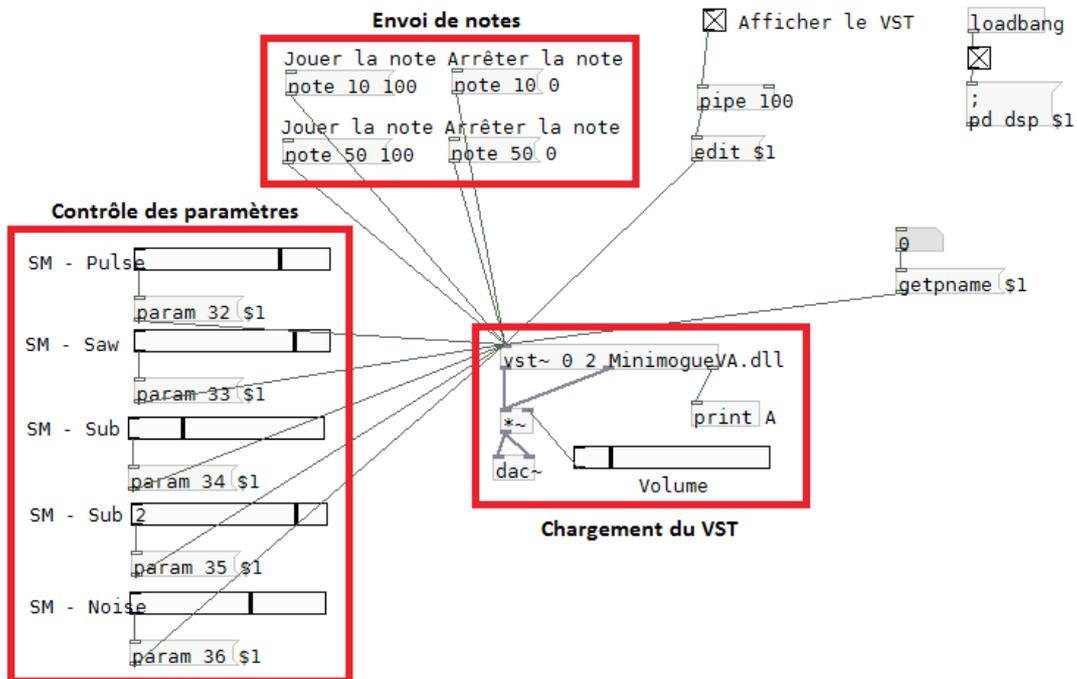
Interaction PureData / VST

Une série de test a été effectuée dans le but de permettre le contrôle d'un VST (en l'occurrence le PolyKB II) depuis PureData. Ces tests nous ont révélé que ce VST ne pouvait pas être contrôlé par MIDI dans sa version gratuite. Plutôt que d'acheter une licence nous avons préféré chercher un autre VST qui serait gratuit et proche du PolyKB II dans ses sonorités et ses caractéristiques. Nous avons rapidement trouvé un synthétiseur correspondant à nos critères : le MinimogueVA.



Interface du MinimogueVA

Les sons émis par cette copie du célèbre Minimoog se rapprochent très fortement de l'esthétique du Syhamo et possèdent globalement les mêmes paramètres. Ce VST a l'avantage (indispensable) de pouvoir être piloté par MIDI, ce que nous avons pu tester à travers un patch PureData.



Patch PureData d'interaction PureData/VST

Nous pouvons distinguer trois zones importantes :

- Une zone de chargement du VST avec la sortie son et le contrôle du volume
- Une zone d'envoi et d'arrêt des notes
- Une zone de contrôle des paramètres

A noter l'utilisation de l'objet `vst~` qui est en fait une librairie externe permettant la communication entre PureData et un VST compatible; librairie disponible uniquement sous Windows.

Systeme quadriphonique

Pour bénéficier de la quadriphonie sur l'ordinateur destiné à faire tourner l'installation, nous avons tout d'abord pensé y brancher une carte son quadriphonique. Cependant, la seule disponible au département, la MOTU 828mk3 ne dispose que d'une connectique FireWire, fréquente sur les Mac mais très rare sur les PC.

Une première solution consisterait à utiliser deux simples cartes sons stéréo, connectées en USB, simultanément. Ce n'est pas nativement possible sous Windows mais Jean-François Charles a suggéré que le serveur audio Jack devrait permettre ce routage particulier.

La seconde solution, dans le cas où nous avons un peu plus de moyens pour l'exposition (voir plus loin) serait de se procurer une carte audio quadriphonique dotée d'une connectique USB.

Design software

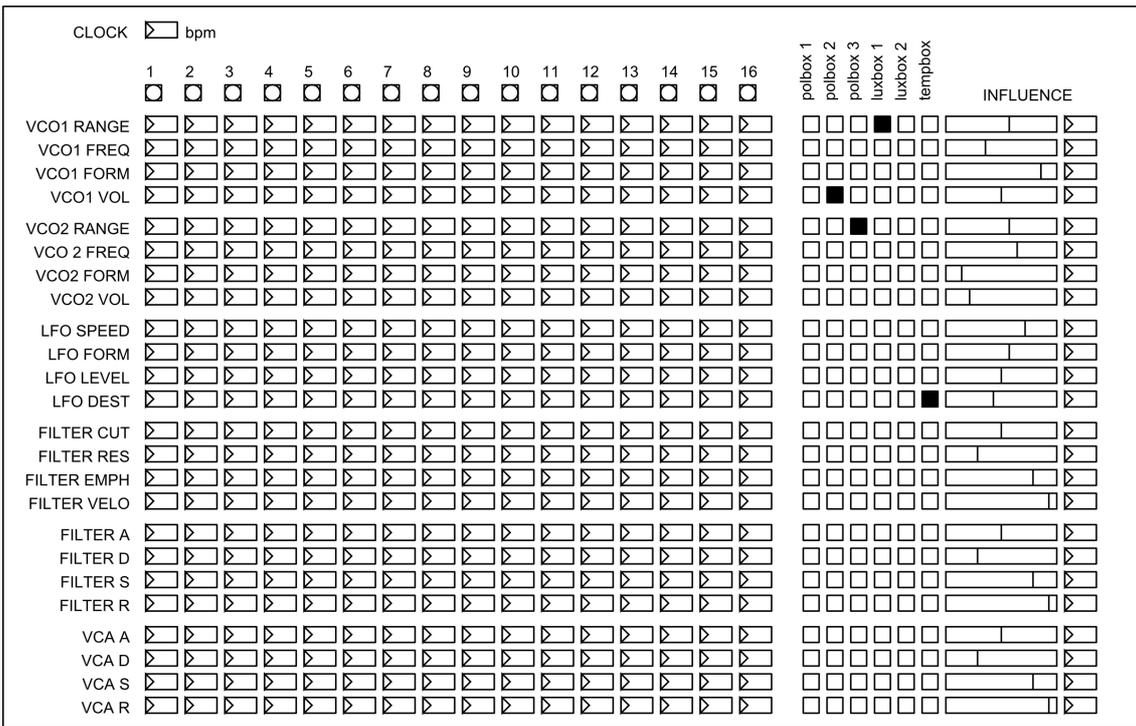
Le spectateur-acteur de l'installation n'a pas besoin de voir la partie logicielle de l'installation, il n'interagit qu'avec les capteurs. Cependant, pour permettre la création de compositions, il est nécessaire de mettre en place une interface logicielle dédiée.

Cette interface doit tout d'abord permettre d'enregistrer les paramètres des 16 sons disponibles sur chaque programmeur. Sur le MinimogueVA, nous avons identifié 24 paramètres susceptibles d'être utilisés (contre 16 sur les Kobol de Wiggle). Ce sont donc 384 nombres qui doivent pouvoir être saisis par les compositeurs, ce qui implique un petit effort ergonomique.

Nous avons choisi de présenter ces informations à la manière d'un séquenceur, avec les 16 sons disposés de gauche à droite. Les paramètres de synthèse sont groupés, pour en faciliter le repérage. Un "bang" au-dessus de chaque son permet de le tester pendant la phase de composition.

Il fallait ensuite mettre en place une interface permettant de relier ("patcher") les capteurs aux paramètres de synthèse qu'ils doivent influencer. L'utilisation de "câbles" PureData aurait été possible et fidèle à l'original, mais des câblages un peu complexes auraient vite été illisibles.

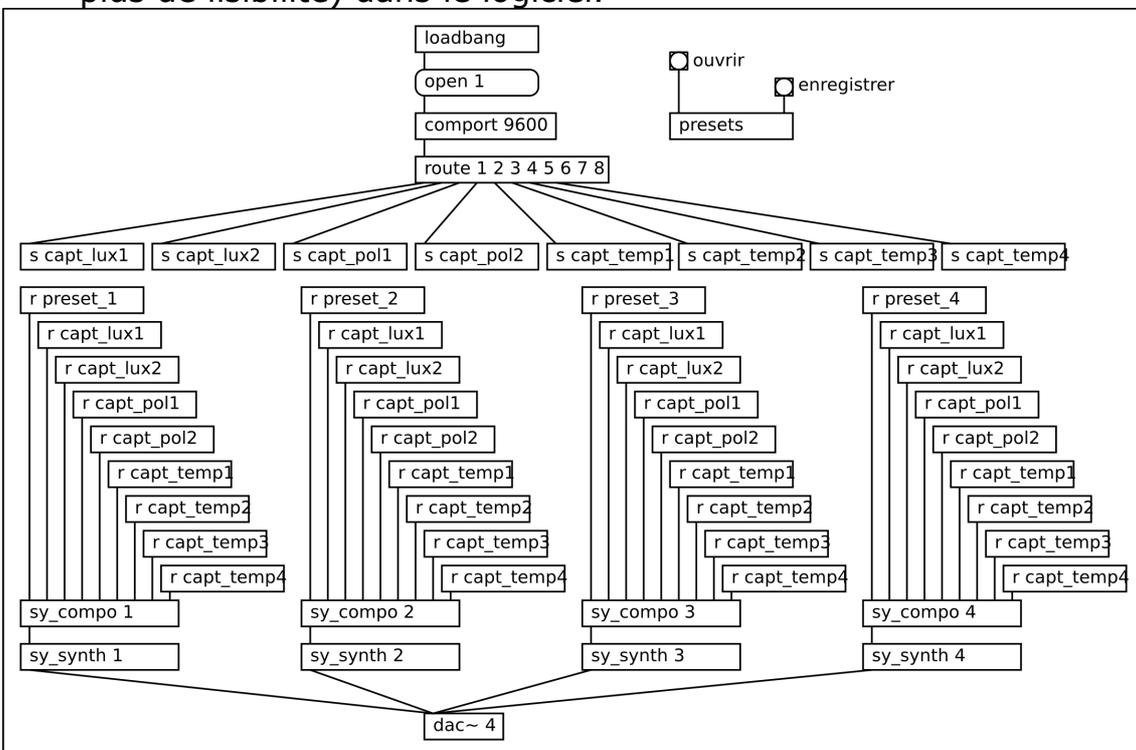
En nous inspirant du synthétiseurs VS3 d'EMS, nous avons donc décidé d'utiliser une matrice pour effectuer ce "patchage". Il suffit de cocher la case à l'intersection d'un capteur et d'un paramètre pour que la liaison soit faite. En plus de son affordance évidente, cette méthode laisse l'interface très lisible, quelle que soit la complexité du patch.



Dessin du séquenceur avec la matrice

A côté de chaque "ligne" de la matrice, un curseur permet de doser l'influence du capteur sur le paramètre.

Enfin, n'oublions pas que cette interface, propre à un programmeur, est disponible en quatre exemplaires (sous forme de sous-patch, pour plus de lisibilité) dans le logiciel.



Dessin du patch général, avec ses 4 objets "compositeur"

Systeme de presets

Contrairement à Max, PureData ne permet pas nativement de sauvegarder l'état (valeurs numérique, boutons radio) d'un patch (ce qu'on appellera ici un preset). Cette fonctionnalité est pourtant nécessaire à l'enregistrement et à la réutilisation de compositions.

Deux principales bibliothèques offrant cette fonctionnalité ont été développées par la communauté de PureData. Nous les avons testées de manière à déterminer laquelle correspondait le mieux à nos besoins.

Ezpst est un projet récent, toujours en développement. Il offre une interface agréable pour enregistrer et charger des presets. Cependant, il ne permet pas de stocker des données complexes (tableaux, listes de nombres), mais seulement des nombres simples. Dans notre cas, l'implémentation serait beaucoup trop complexe. Par ailleurs, il ne propose pas de choisir l'emplacement où sauvegarder les presets (bien que cela soit modifiable dans le code). Enfin, et c'est ce qui rend ces défauts rédhibitoires, il ne semble que très peu utilisé, et les informations disponibles en ligne sont extrêmement rares.

Sssad, bien que toujours en version bêta, est beaucoup plus ancien (il a même, un temps, fait partie de pd-extended). La communauté l'a donc très largement utilisé, documenté et amélioré. Une interface avancée de sauvegarde/chargement permet de manipuler les presets (sous forme de fichiers textes) très finement. Surtout, il permet de stocker tout type de données utilisées par PureData. Dans notre cas, cela permet de stocker directement des listes de nombres, et donc de simplifier l'implémentation.

C'est donc **Sssad** que nous avons retenu pour notre programme.

Installation

Disposition optimale du Sy~

Sy~ doit être disposé dans une pièce suffisamment vaste pour permettre d'utiliser toute l'amplitude de mesure des capteurs de distance (7 mètre en face de l'installation). Un éclairage à la verticale du boîtier permet de maximiser le fonctionnement des capteurs de lumière (voir réglage plus loin). Les capteurs de température ne demandent pas de conditions particulières mais des températures extrêmes ($< 5^{\circ}\text{C}$ ou $> 53^{\circ}\text{C}$) peuvent perturber leur fonctionnement (et, plus généralement, celui de l'ensemble du circuit).

L'espace dédié à l'installation doit être équipé d'une sonorisation quadraphonique (une enceinte à chaque coin de la pièce, par exemple). La qualité de l'expérience musicale dépendra fortement de celle des enceintes. Enfin la réverbération naturelle de la salle ne doit pas être trop importante, pour rester fidèle à la composition (qui comprend déjà un système de réverbération logicielle).

Schéma d'installation

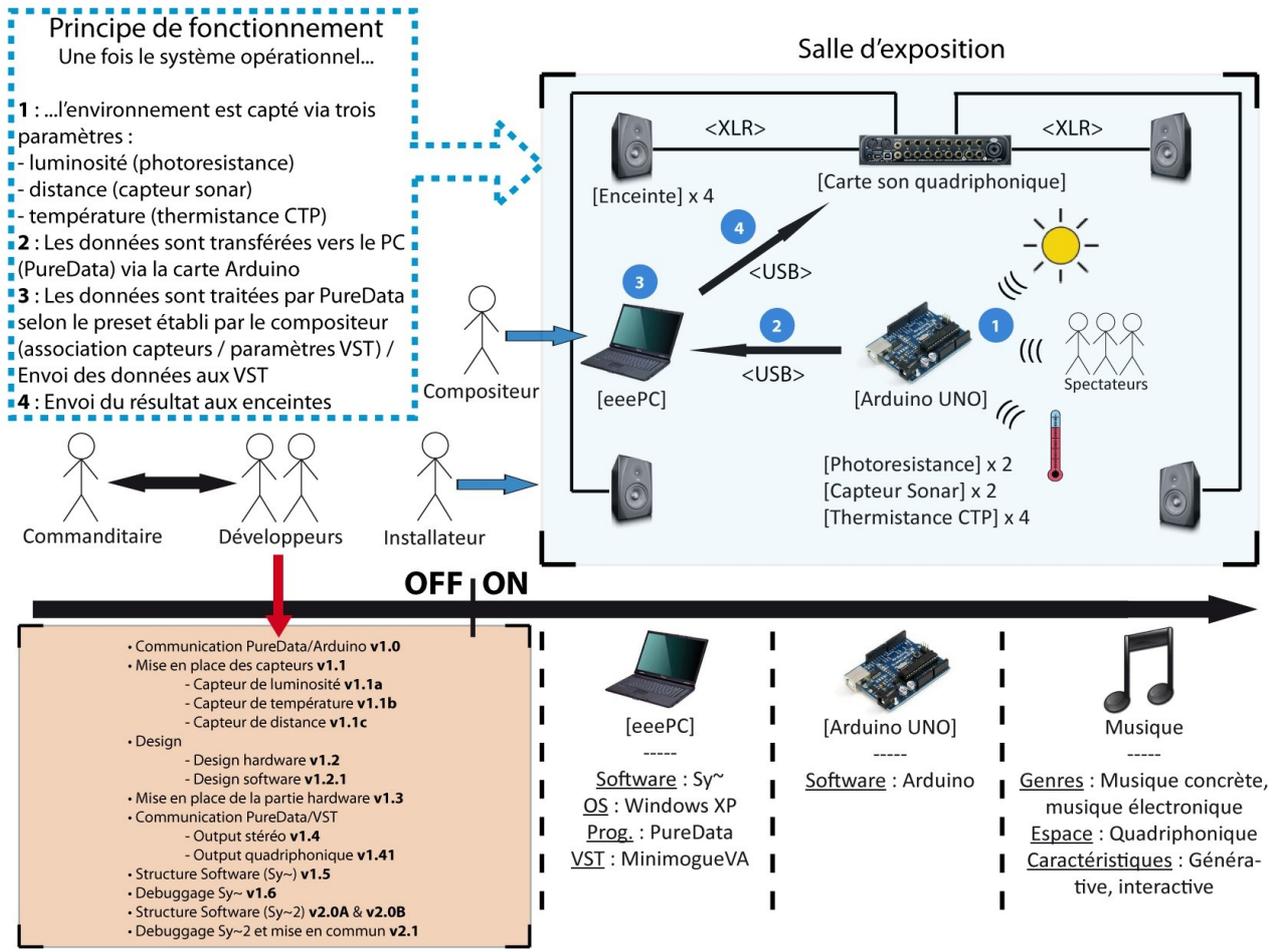


Schéma d'installation du Sy~

Principe d'étalonnage de la luminosité

Afin d'optimiser la captation de la luminosité (si le Sy~ est installé dans un environnement naturellement sombre, il faut que le logiciel puisse tout de même enregistrer des variations importantes entre le noir et la luminosité ambiante) nous avons imaginé un système

d'étalonnage de celle-ci. Un bouton poussoir permettra d'enregistrer la valeur de luminosité maximale (sur une dizaine de secondes, et en prévoyant une marge de sécurité), et les valeurs mesurées seront quantifiées en fonction de ce maximum.

Mise en route

La mise en route de l'installation se fera automatiquement après la mise sous tension de l'ordinateur, la carte Arduino ainsi que tous les composants électriques étant alimentés directement par USB. Un script batch (.bat) sera nécessaire afin de lancer directement le patch Pd, qui, grâce au système de preset, sera automatiquement opérationnel.

Posters et expositions

Une première opportunité d'exposition nous a été offerte par le Conservatoire de Montbéliard, dès le début du projet. Sy~ pourrait être installé dans la bibliothèque du Conservatoire (ou dans une pièce dédiée), à disposition des visiteurs.

Plus récemment, nous avons été invités à participer à une exposition à Shanghai, organisée par Pays de Montbéliard Agglomération, sous la direction artistique de Jacopo Baboni Schilingi (qui intervient dans le parcours Musique et Son). Si cette piste devait se confirmer, du matériel dédié à l'installation (ordinateur et carte son) devrait être acheté, et le circuit électronique devrait être construit en deux exemplaires (pour éviter les pannes).

Dans ces deux cas, Jean-François Charles a suggéré la création de posters pour présenter notre travail sur le lieu d'exposition (en anglais pour Shanghai). Un poster pourrait présenter le travail original d'Oscar et Janine Wiggli, et le second notre démarche de création.

Gestion de projet

Suite aux recherches pour le cahier de spécification, certains points du projet ont été abordés plus tôt que prévu et nous avons ainsi pu prendre de l'avance dans la production. A l'inverse nous pensions recevoir les capteurs avant les vacances (et faire les tests pendant celles-ci), ce qui n'a pas été le cas.

C'est pourquoi nous intégrons dans ce dossier la nouvelle version du diagramme qui est, à présent, accompagné du tableau des tâches et des responsabilités.

Tableau des tâches

Tâche / Sous-tâche	Temps	Livrable	Resp.
Recherches	-	-	Antonin
Bibliographie / Etude des ouvrages	-	-	Cédric
Test de communication PureData / Arduino	-	-	Cédric
Recherche d'un VST de Kobol ou création d'un	-	-	Antonin
Définition des composants nécessaires et du câblage	-	-	Antonin
Compte-rendus hebdomadaires	1j/CR	Compte-rendus	Cédric
Cahier des charges	20j	Cahier des charges	Cédric
Etablir une charte graphique pour l'ensemble des livrables	6j	-	Antonin
Création de l'identité visuelle	6j	-	Antonin
Rédaction	16j	-	Cédric
Mise en page	3j	-	Antonin
Rendu	Jalon	Cahier des charges	Cédric

Cahier de spécification	20j	Cahier de spécification	Cédric
Test et recherches	23j	-	Antonin
Rédaction	23j	-	Cédric
Mise en page	2j	-	Antonin
Rendu	Jalon	Cahier de spécification	Cédric
Communication Pd/Arduino	10j	Prototype v1.0	Antonin
Mise en place des capteurs	10j	Prototype v1.1	Cédric
Capteur luminosité	-	- v1.1a	Cédric
Capteur température	-	- v1.1b	Cédric
Capteur distance	-	- v1.1c	Antonin
Design	5j		Antonin
Design Hardware	3j	Prototype v1.2	Cédric
Design Software	2j	Prototype v1.2.1	Antonin
Mise en place de la partie hardware	15j	Prototype v1.3	Cédric
Communication Pd/VST	15j	-	Cédric
Output stéréo/mono	8j	Prototype v1.4	Cédric
Output quadraphonique	7j	Prototype v1.4.1	Antonin
Structure Software Sy~	15j	Prototype v1.5	Antonin
Debuggage Sy~	5j	Prototype v1.6	Antonin
Structure Software Sy~ 2	20j	Prototype v2.0A & v2.0B	Cédric / Antonin
Debuggage Sy~ 2 et mise en commun	5j	Prototype v2.1	Cédric
Rapport final	10j	Rapports finaux	Cédric
Rédaction	7j	-	Cédric
Mise en page	2j	-	Antonin
Rendu	Jalon	Rapport final	Cédric
Rendu print	Jalon	Rapport final (print)	Antonin
Préparation soutenance	3j	-	Antonin
Soutenance finale	Jalon	-	Antonin

Diagramme de Gantt

