

FORMAT DE TRANSMISSION DE DONNÉES OPEN SOUND CONTROL

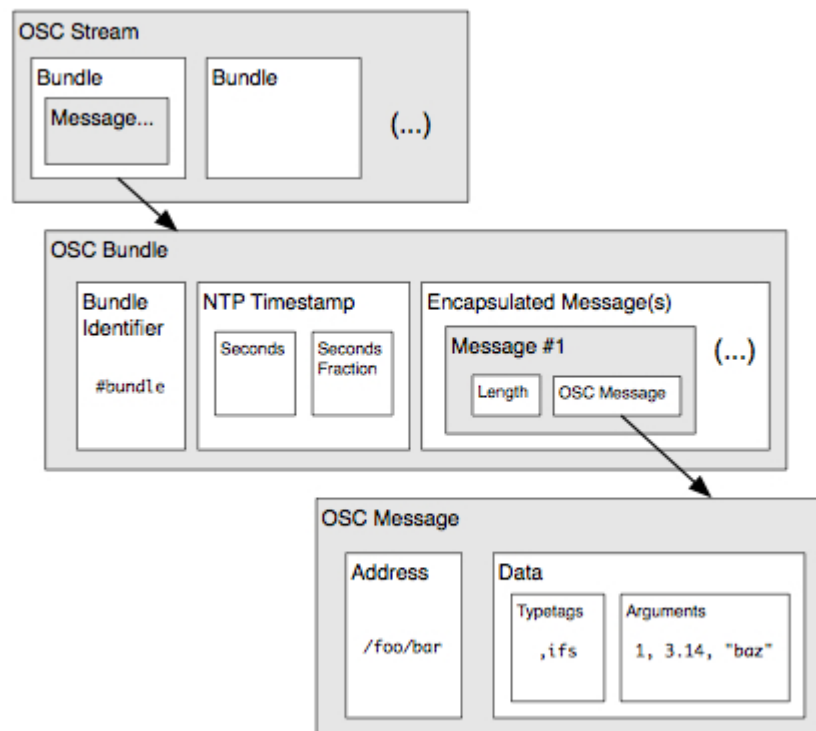
L'**Open Sound Control** est un format de transmission de données entre ordinateurs, synthétiseurs, robots ou tout autre matériel ou logiciel compatible, conçu pour le **contrôle en temps réel**. Il utilise le réseau au travers des protocoles UDP ou TCP et apporte des améliorations en termes de rapidité et flexibilité par rapport à l'ancienne norme MIDI.

Source : Wikipedia

Ce format est le plus répandu actuellement dans les logiciels de musique et dans les arts numériques. Il est considéré comme le successeur du **MIDI**.

Le protocole UDP sera préféré pour une question de rapidité et de fluidité de la transmission d'informations. Mais il faut avoir conscience que des données peuvent se perdre en cours de route ou être envoyées plusieurs fois.

Architecture du protocole OSC



Un flux OSC est composé de plusieurs **paquets** (Bundle).

Un **paquet** est composé des champs suivants :

- un numéro identifiant
- un champ qui indique l'horodatage du paquet : horodatage au format NTP (nombre de secondes écoulées depuis le 1er janvier 1900 minuit) et le nombre de fractions de secondes (précision de 200 picosecondes)
- un ou plusieurs message(s) encapsulé(s).

Un **message encapsulé** est composé des champs suivants :

- la taille du message OSC
- le message OSC

Un **message OSC** est composé des champs suivants :

- l'adresse de destination : c'est une chaîne de caractères qui commence par un /. Cette chaîne de caractères suit un schéma d'arborescence de répertoires.
par exemple : /voices/synth1/osc1/modfreq.
- Le(s) type(s) de données des arguments transmis :
 - le protocole OSC 1.0 admet 5 types de données :
 - i : entier 32 bits (format big endian complément à 2)
 - t : balise temps 64 bits (big endian virgule fixe)
 - f : réel 32 bits (format big endian virgule flottante)
 - s : chaîne de caractères
 - b : blob (binary large object, regroupement de plusieurs données binaires en une seule entité ; exemple des données audio, une image, une image vidéo)
- Les valeurs des arguments

Dans la figure précédente, l'exemple suivant est donné :

- adresse OSC : /foo/bar
- message OSC : ,ifs 1, 3.14, "baz"

Le message OSC contient 3 arguments, le premier est un entier et sa valeur est égale à 1, le deuxième est un réel et sa valeur vaut 3.14 et le troisième est une chaîne de caractères et sa valeur vaut baz.

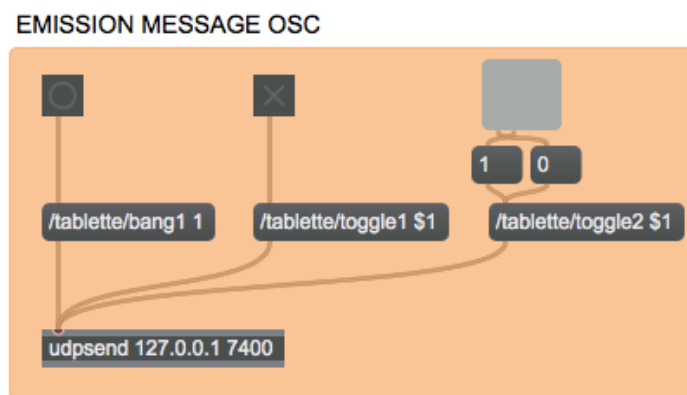
Mise en œuvre avec MAX

Nous utiliserons les modules OSC de la bibliothèque CNMAT. Avant toute chose, il faut ajouter cette bibliothèque dans vos préférences de MAX : File → Show Package Manager → CNMAT Externals → installer

Si les objets MAX de la bibliothèque CNMAT Externals apparaissent en orange au lieu d'être en gris, cela signifie que la bibliothèque n'a pas été chargée.

Mise en oeuvre de l'envoi de données OSC par MAX

Voici un exemple de mise en œuvre du protocole OSC avec MAX sur une liaison UDP pour l'envoi de données :



Il faut utiliser l'objet **[udpsend]**. Cet objet possède deux arguments :

- Le premier argument est l'adresse IP du destinataire des données : dans notre exemple 127.0.0.1,
- Le deuxième argument correspond au numéro de port utilisé : dans notre exemple 7400.

Ensuite pour envoyer au destinataire un message OSC, il suffit d'envoyer un message qui respecte le format OSC à savoir une adresse OSC suivie des données qu'on souhaite envoyer. Dans l'exemple donné, nous avons trois messages OSC possibles :

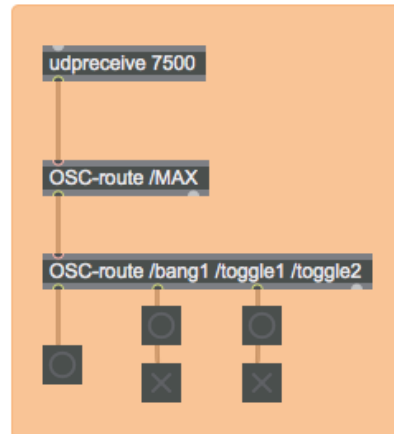
- Dès qu'on appuie sur le bang, on envoie la donnée 1 à l'adresse OSC suivante /tablette/bang1,
- Dès qu'on appuie sur l'interrupteur, on envoie l'état de l'interrupteur à l'adresse OSC suivante /tablette/toggle1,
- Le troisième élément est une nouvelle interface. Il s'agit d'un [ubutton]. Il permet de détecter les événements mousedown et mouseup. Dès qu'on clique sur le bouton gauche de la souris (sans relâcher) on envoie la donnée 1 à l'adresse OSC suivante /tablette/toggle2 et dès qu'on relâche le bouton gauche de la souris on envoie la donnée 0 à l'adresse OSC suivante /tablette/toggle2.

Pour le reste (mise en conformité par rapport au format OSC) tout est géré automatiquement par l'objet [udpsend], ce qui simplifie grandement la mise en oeuvre.

Mise en oeuvre de la réception de données OSC par MAX

Voici un exemple de mise en œuvre du protocole OSC avec MAX sur une liaison UDP pour la réception de données :

RECEPTION MESSAGE OSC



Pour recevoir les messages OSC, il faut utiliser l'objet **[udpreceive]**. Cet objet possède un argument :

- L'argument correspond au numéro de port écouté : dans notre exemple 7500.

En sortie de cet objet, nous récupérons directement les adresses OSC suivies des données. Si nous affichons à l'aide d'un objet print les messages en sortie de l'objet udpsend dans notre exemple nous pourrions obtenir cela :

- /MAX/toggle1 1.
- /MAX/toggle1 1.
- /MAX/toggle2 1.
- /MAX/bang1 1.
- /MAX/bang1 1.

Ensuite pour récupérer les bonnes données dans le bon bout de code MAX, il faut utiliser des objets [OSC-route]. Cet objet fonctionne comme l'objet [route], sauf qu'il traite des adresses OSC. L'argument de OSC-route permet justement de détecter l'adresse OSC souhaitée.

Par exemple l'objet [route-OSC /MAX] va récupérer tous les messages OSC dont l'adresse OSC commence par /MAX. Les messages ayant cette caractéristique seront renvoyés par la première sortie (celle tout à gauche) sans le terme MAX afin de poursuivre le traitement. Les messages n'ayant pas cette caractéristique seront renvoyés par la sortie tout à droite, telle qu'elle. Si nous reprenons la liste précédente de messages envoyés, tous ces messages commencent par /MAX, donc on aura la liste suivante de messages qui sortiront de la sortie tout à gauche :

- /toggle1 1.
- /toggle1 1.
- /toggle2 1.
- /bang1 1.
- /bang1 1.

Ensuite le deuxième objet [OSC-route /bang1 /toggle1 /toggle2] va permettre de traiter la deuxième partie du message OSC. Les messages ayant /bang1 comme deuxième partie d'adresse OSC vont sortir par la sortie tout à gauche. Les messages ayant /toggle1 comme deuxième partie d'adresse OSC vont sortir par la deuxième sortie à partir de la gauche. Les messages ayant /toggle2 comme deuxième partie d'adresse OSC vont sortir par la troisième sortie à partir de la gauche. Les messages qui ne commencent ni par /bang1, ni par /toggle1 et ni par /toggle2 vont sortir par la sortie tout à droite.

Sources du code :

OSC côté MAX :

<http://cnmat.berkeley.edu/downloads>

Sources Informations de ce document :

Wikipedia : article [Open Sound Control](#)

Spécifications OpenSound Control : http://opensoundcontrol.org/spec-1_0