

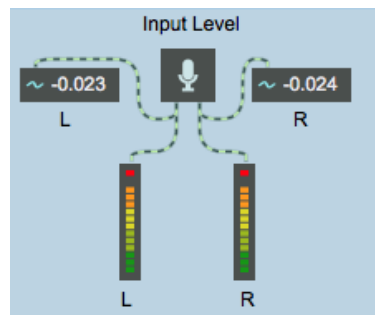
40-signaux et vu-mètres

Ce chapitre présente plusieurs objets MSP permettant d'observer la valeur numérique des signaux et de convertir ces valeurs en messages Max.

Afficher la valeur d'un signal: *number ~*

- Activez l'audio et envoyez du son dans les jack d'entrée de l'ordinateur.

Toutes les 250 millisecondes, les objets *number ~* en haut du Patcher affichent la valeur actuelle du signal entrant dans chaque canal, et les objets *meter ~* affichent une représentation graphique de la valeur maximale de l'amplitude au cours des 250 dernières millisecondes, à la manière d'un affichage analogique à LED.

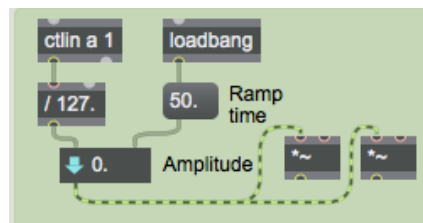


La valeur actuelle du signal est indiquée par *number ~*; l'amplitude maximale est indiquée par *meter ~*

Le signal entrant dans *number~* est envoyé par sa sortie droite comme un **float** une fois à chaque fois qu'il est affiché. Cela signifie qu'il est possible d'échantillonner la valeur du signal et de l'envoyer comme message à d'autres objets Max.

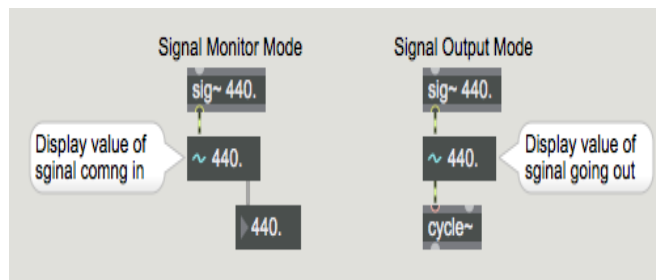
L'objet *number ~* est en fait deux objets en un. En plus de recevoir les valeurs de signal et de les envoyer par la sortie de droite sous forme de **float**, *number~* fonctionne également comme une boîte de *nombres* à virgule flottante qui envoie un **signal** (au lieu d'un **float**) par la sortie de gauche.

- Déplacez la molette de votre clavier MIDI ou faites glisser le curseur sur le côté droit du *number~* marqué «Amplitude». Ceci définit la valeur du signal envoyé à la sortie gauche de l'objet *number~*. Le signal est connecté à l'entrée droite de deux objets **~*, pour contrôler l'amplitude du signal envoyé à *ezdac~*.



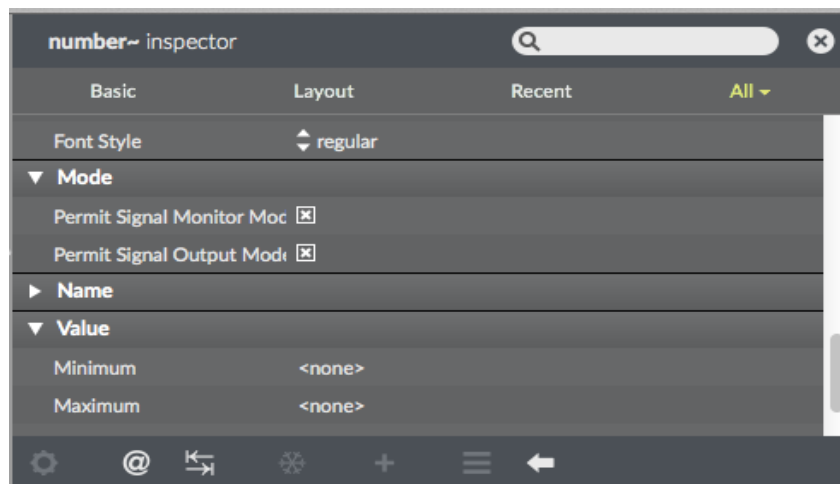
float input to *number ~* définit la valeur du signal envoyé à la sortie gauche

Un objet *number~* convertit simultanément tout signal reçu en **floats** envoyés par la sortie droite et convertit tout **float** reçu en un signal envoyé par la sortie gauche. Bien qu'il puisse effectuer ces deux tâches en même temps, il ne peut afficher qu'une seule valeur à la fois. La valeur affichée par *number~* dépend du mode d'affichage utilisé. Lorsqu'une petite forme d'onde apparaît dans la partie gauche de *number~*, il est en **mode de sortie du signal** et indique la valeur du signal sortant par la sortie gauche.



Les deux modes d'affichage de *number~*

Vous pouvez restreindre le *number~* à l'un ou l'autre mode d'affichage en sélectionnant l'objet dans un Patcher déverrouillé, et en choisissant **Lire les informations ...** dans le menu Objet.



Les modes d'affichage autorisés peuvent être choisis dans l'Inspecteur de *number~*

Au moins un mode d'affichage doit être coché. Par défaut, les deux modes d'affichage sont autorisés, comme le montre l'exemple ci-dessus. Si les deux modes d'affichage sont autorisés, vous pouvez passer d'un mode d'affichage à l'autre dans un Patcher verrouillé en cliquant sur le côté gauche de *number~*. La sortie de *number~* se poursuit quel que soit le mode d'affichage dans lequel il se trouve.

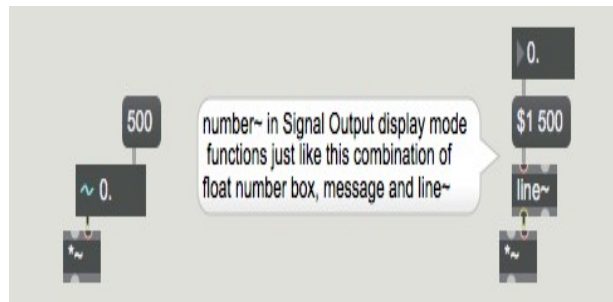
Dans le patch du didacticiel, vous pouvez voir les deux modes d'affichage de *number~*. Les objets *number~* en haut de la fenêtre du Patcher sont en mode **Moniteur de signal** car nous les utilisons pour afficher la valeur du signal entrant. Le *number~* «Amplitude» ~ est en mode de **sortie du signal** car nous l'utilisons pour envoyer un signal et nous voulons voir la valeur de ce signal. (De nouvelles valeurs peuvent être saisies dans un *number~* en les tapant ou en les faisant glisser avec la souris uniquement lorsqu'il est en mode d'affichage **Signal de sortie**.) Chacun de ces objets *number~* ne remplit qu'une seule fonction, ils ont été limités à un seul mode d'affichage dans la fenêtre Inspecteur.

- Cliquez sur le côté gauche des objets *number~*. Ils ne changent pas de mode d'affichage car ils ont été limités à l'un mode ou l'un autre dans la fenêtre de l'inspecteur.

Interpolation avec *number~*

L'objet *number~* possède une fonctionnalité supplémentaire très utile. Il peut être utilisé pour interpoler entre les valeurs d'entrée afin de générer un signal de rampe semblable à l'objet *line~*. Si *number~* reçoit un nombre non nul dans son entrée droite, il utilise ce nombre comme une quantité

de temps, en millisecondes, pour interpoler linéairement à nouvelle valeur chaque fois qu'il reçoit un nombre dans l'entrée gauche. Cela équivaut à l'envoi d'une liste à *line~*.

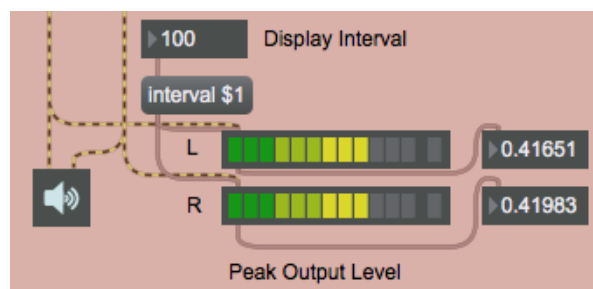


number~ peut envoyer un signal de rampe linéaire de son ancienne valeur à une nouvelle valeur

Contrairement à *lin~*, toutefois, *number~* n'a pas besoin de recevoir la valeur du temps d'interpolation plus d'une fois; il se souvient du temps d'interpolation et l'utilise pour chaque nouveau nombre reçu dans l'entrée gauche. Cette caractéristique est utilisée pour le *number~* «Amplitude» afin qu'il ne provoque pas de changements discontinus d'amplitude dans le signal de sortie.

Amplitude de crête: *meter~*

L'objet *meter~* affiche périodiquement l'amplitude de crête qu'il a reçue depuis le dernier affichage. En même temps, il envoie également la valeur du signal de crête à sa sortie sous forme de **float**. La valeur de sortie est toujours un nombre positif, même si la valeur de crête était négative.



meter~ affiche l'amplitude du signal de crête et l'envoie sous forme de float

meter~ est utile pour observer l'amplitude maximale d'un signal (contrairement à *number~*, qui affiche et envoie l'amplitude instantanée du signal). Étant donné que *meter~* est destiné aux signaux audio, il s'attend à recevoir un signal compris entre -1 et 1. Si cette plage est dépassée, *meter~* affiche une LED rouge d'écrêtage comme maximum.

- Si vous souhaitez voir l'affichage de l'écrêtage, augmentez l'amplitude du signal de sortie jusqu'à ce qu'il dépasse 1. (Ramenez-le ensuite à un niveau souhaitable).

L'intervalle de temps par défaut entre les mises à jour d'affichage de *meter~* est de 250 millisecondes, mais l'intervalle d'affichage peut être modifié avec le message **interval**. Un intervalle d'affichage plus court rend l'affichage LED plus précis, tandis qu'un intervalle plus long vous donne plus de temps pour lire sa sortie visuelle et numérique.

- Vous pouvez essayer différents intervalles d'affichage en modifiant le nombre dans la boîte de *nombre* «Intervalle d'affichage» située dans le coin inférieur gauche de la fenêtre du Patcher.

A propos, l'intervalle d'affichage d'un objet *meter~* peut être défini de la même manière (ainsi que via sa fenêtre Inspecteur).

Détection du signal

Vous n'êtes pas obligé d'inclure les objets *number~* et *meter~* dans votre patch, sauf s'ils sont nécessaires pour l'interface utilisateur. Vous pouvez mesurer n'importe quel signal audio à tout moment avec la fonction de détection de signal.

- Cochez **Probing** dans le menu **Débogage**

Maintenant, lorsque l'audio est activé, vous pouvez pointer la souris sur un patch audio et voir une bulle avec une mesure du signal et un affichage du vu-mètre. Cela vous permet de vérifier la santé du signal n'importe où dans le patch.



Détection de signaux

Utilisez un signal pour générer des messages Max: snapshot ~

L'objet *snapshot~* envoie la valeur actuelle d'un signal, tout comme l'entrée droite de *number~*. Avec *snapshot~*, cependant, vous pouvez activer et désactiver la sortie, ou demander la sortie d'une seule valeur en lui envoyant un **bang**. Lorsque vous envoyez un nombre non nul dans l'entrée droite, *snapshot~* utilise ce nombre comme un intervalle de temps en millisecondes et commence à rapporter périodiquement la valeur du signal dans son entrée gauche. L'envoi d'un intervalle de temps de **0** arrête *snapshot~*.

Cette moitié droite du patch du didacticiel montre un exemple simple de la façon dont une forme d'onde de signal peut être utilisée pour générer des données MIDI. Nous allons échantillonner une onde cosinusoidale sub-audio pour obtenir des valeurs de hauteur pour les messages de note MIDI.

- Utilisez le *number~* pour régler l'amplitude de sortie sur **0**. Dans les objets de la boîte de *nombre* en haut du patch, réglez la boîte de nombre «Rate» sur **0,14** et réglez la boîte de nombre «Depth» sur **0,5**. Cliquez sur la boîte de *message 200* pour que *snapshot~* commence à rapporter les valeurs du signal tous les cinquièmes de seconde.

Comme *snapshot~* rapporte la valeur du signal tous les cinquièmes de seconde et que la période de l'objet *cycle~* est d'environ 7 secondes, la mélodie décrira un cycle d'une onde sinusoïdale toutes les 35 notes. L'amplitude de l'onde étant de 0,5, la mélodie sera comprise entre 36 et 84 (60 ± 24).

- Expérimentez avec différentes valeurs de «Rate» et de «Depth» pour l'objet *cycle~*. Comme *snapshot~* échantillonne à une fréquence de 5 Hz (une fois toutes les 200 ms), sa fréquence de Nyquist est de 2,5 Hz, ce qui limite la fréquence effective de *cycle~* (toute fréquence supérieure sera «repliée»). Cliquez sur boîte de *message 0* pour arrêter le *snapshot~*.

Modulation de l'amplitude

- Réglez la profondeur du trémolo sur **0,5** et le taux de trémolo sur **4**. Augmentez l'amplitude de sortie à un niveau d'écoute souhaitable.

L'objet *cycle* ~ module l'amplitude du son entrant avec un tremolo de 4 Hz.

- Expérimentez des taux de modulation plus rapides (gamme audio) pour entendre l'effet de timbre de la modulation d'amplitude. Pour entendre la modulation en anneau, définissez la profondeur de modulation sur **1**. Pour supprimer l'effet de modulation, réglez simplement la profondeur sur **0**.

Visualiser un extrait de signal: *capture* ~

L'objet *capture* ~ est comparable à l'objet Max *capture*. Il stocke de nombreuses valeurs de signal (les 4096 échantillons les plus récemment reçus, par défaut), de sorte que vous pouvez visualiser un extrait entier d'un signal sous forme de texte.

- Réglez la profondeur du trémolo sur **1** et le taux de trémolo sur **172**. Double-cliquez sur l'objet *capture* ~ pour ouvrir une fenêtre de texte contenant les 4096 derniers échantillons.

Cet objet est utile pour voir précisément ce qui s'est passé dans un signal au fil du temps. (4096 échantillons représentent environ 93 millisecondes à une fréquence d'échantillonnage de 44,1 kHz.) Vous pouvez entrer un argument pour spécifier le nombre d'échantillons que vous souhaitez visualiser, et *capture*~ stockera ce nombre d'échantillons (en supposant que suffisamment de RAM soit disponible dans Max.) Il existe plusieurs arguments et messages pour contrôler exactement ce qui sera stocké par *capture*~. Voir sa description dans le Manuel de référence du MSP pour plus de détails.

Résumé

L'objet *capture*~ stocke un court extrait d'un signal pour le visualiser sous forme de texte. L'objet *meter*~ affiche périodiquement le niveau de crête d'un signal et envoie le niveau de crête à sa sortie sous forme de float. L'objet *snapshot*~ envoie un float pour rapporter la valeur actuelle d'un signal. *snapshot*~ rapporte la valeur du signal une fois qu'il reçoit un **bang**, et il peut également rapporter la valeur du signal périodiquement s'il reçoit un intervalle non nul dans son entrée droite.

L'objet *number*~ est comme une combinaison d'une boîte de *nombres float*, d'un *sig*~ et d'un *snapshot*~, tout à la fois. Un signal reçu dans son entrée gauche est envoyé par la sortie droite sous forme de **float**, comme avec un *snapshot*~. Un **float** ou **int** reçu dans son entrée gauche définit la valeur du signal qui sort par sa sortie gauche, comme avec *sig*~. Ces deux activités peuvent se dérouler simultanément dans le même objet *number*~, bien que *number*~ ne puisse afficher qu'une valeur à la fois. Lorsque *number*~ est en mode de sortie de signal, il affiche la valeur du signal sortant.

number~ peut également fonctionner en tant que générateur de rampe de signaux, comme l'objet *line*~. Si un nombre non nul a été reçu dans l'entrée droite (représentant le temps d'interpolation en millisecondes), chaque fois que *number*~ reçoit un **float**, son signal de sortie interpole linéairement entre l'ancienne et la nouvelle valeur.

Ces objets (ainsi que quelques autres, tels que *sig~floating-point number box* et *avg* ~) constituent les principaux liens entre MSP et Max. Ils convertissent les signaux en messages numériques Max, ou vice versa.